SIMATIC Cartesian Portal

SIMATIC S7 1500

https://support.industry.siemens.com/cs/ww/en/view/109815631

Siemens Industry Online Support

Legal information

Use of application examples

Application examples illustrate the solution of automation tasks through an interaction of several components in the form of text, graphics and/or software modules. The application examples are a free service by Siemens AG and/or a subsidiary of Siemens AG ("Siemens"). They are non-binding and make no claim to completeness or functionality regarding configuration and equipment. The application examples merely offer help with typical tasks; they do not constitute customer-specific solutions. You yourself are responsible for the proper and safe operation of the products in accordance with applicable regulations and must also check the function of the respective application example and customize it for your system.

Siemens grants you the non-exclusive, non-sublicensable and non-transferable right to have the application examples used by technically trained personnel. Any change to the application examples is your responsibility. Sharing the application examples with third parties or copying the application examples or excerpts thereof is permitted only in combination with your own products. The application examples are not required to undergo the customary tests and quality inspections of a chargeable product; they may have functional and performance defects as well as errors. It is your responsibility to use them in such a manner that any malfunctions that may occur do not result in property damage or injury to persons.

Disclaimer of liability

Siemens shall not assume any liability, for any legal reason whatsoever, including, without limitation, liability for the usability, availability, completeness and freedom from defects of the application examples as well as for related information, configuration and performance data and any damage caused thereby. This shall not apply in cases of mandatory liability, for example under the German Product Liability Act, or in cases of intent, gross negligence, or culpable loss of life, bodily injury or damage to health, non-compliance with a guarantee, fraudulent non-disclosure of a defect, or culpable breach of material contractual obligations. Claims for damages arising from a breach of material contractual obligations shall however be limited to the foreseeable damage typical of the type of agreement, unless liability arises from intent or gross negligence or is based on loss of life, bodily injury or damage to health. The foregoing provisions do not imply any change in the burden of proof to your detriment. You shall indemnify Siemens against existing or future claims of third parties in this connection except where Siemens is mandatorily liable.

By using the application examples you acknowledge that Siemens cannot be held liable for any damage beyond the liability provisions described.

Other information

Siemens reserves the right to make changes to the application examples at any time without notice. In case of discrepancies between the suggestions in the application examples and other Siemens publications such as catalogs, the content of the other documentation shall have precedence.

The Siemens terms of use (https://support.industry.siemens.com) shall also apply.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit https://www.siemens.com/industrialsecurity.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under https://www.siemens.com/cert.

Table of contents

1 Introduction

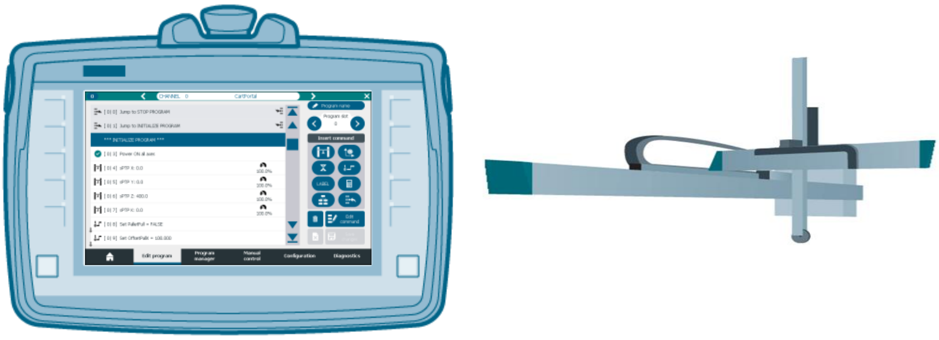
1.1 Overview

SIMATIC Cartesian Portal is a plug-and-play software solution for robot control. It consists of a pre-built PLC project with an HMI application. The operating concept allows the user to create, edit and execute robot programs entirely on the HMI without an engineering system. The SIMATIC Cartesian Portal solution is a software product. Via the Online Software Delivery (OSD) system, one license per controller must be purchased to operate the software.

Following benefits are provided with this solution:

- Create new programs and edit existing programs via HMI
- Flexible program structure labels, branches, loops, subroutines, and parallel sequences
- Support of up to six axes for multi-axis positioning with sPTP
- Usage of independent single axes which are not part of the multi-axis system
- Teach-in positions
- Save programs on HMI or USB flash drive
- Jog the kinematics by single axes or to Cartesian positions
- Configuration of axes parameter e.g., position limits and dynamic limits
- Configuration of inputs, outputs, and variables for usage in the program
- Configuration of several blocked zones to avoid collisions
- Diagnosis of axes status, zones, variables, and errors

Figure 1-1: SIMATIC Cartesian Portal



SIMATIC Cartesian Portal supports the following Kinematics types and movements:

- Cartesian portals with different degrees of freedom
- Synchronous point-to-point movement
- Blending of motion commands
- Single axis positioning for additional axes, e.g., conveyor belts

1.2 Supported functionalities of SIMATIC Cartesian Portal

The SIMATIC Cartesian Portal solution contains a complete HMI project to operate the Kinematics and additional single axes. This consists of several independent faceplates. Each of the faceplates represents a module of combined useful functionalities. The screens with these faceplates are called on the HMI by another faceplate, that represents the main navigation bar. Each faceplate and its purpose is listed in the following.

Home screen

- Initialize the program execution
- Start, Stop and Pause the automatic program
- Single step operation mode for program testing
- Change axes override
- Display of the currently executed command and the following commands
- Use user-defined buttons for custom functionality

Edit program

- The editor allows the user to create the automatic program via the following described functionalities.
- sPTP Motion Control command for the Kinematics with usage of teached points, variables, blending, and Dynamics setting
- Single axis Motion Control commands
- Set outputs or conditions
- Calculate, increment, or assign absolute values to variables
- Jumps and conditional jumps
- Loops, e.g., for subroutines
- Start and stop parallel sequences/programs
- Wait commands for variables, inputs, output, conditions, and time (specified manually or by a variable)
- Labels and comment functions
- Copy, paste, shift of single commands
- Automatic creation of a starting sequence

Program manager

- Save the current PLC program on the HMI or on a USB drive
- Load a saved program or parts of it from the HMI or USB drive to the PLC

Manual control

- Configure or teach-in positions in a point table
- Enable / disable a single axis or all kinematics axes at once
- Acknowledge pending axis errors
- Open / close the motor holding brake
- Jog a single axis incrementally, continuously or to a target position
- Jog the kinematics to a point from the point table with sPTP
- Home axis
- Active homing
- Absolute encoder adjustment
- Direct homing

Configuration

- Configuration of software and hardware position limits and reference values for torque/force limiting
- Configuration of inputs, outputs, conditions, and variables
- Configuration of AND/OR-logic operations to query the signal states of two or more inputs or conditions and save them in a condition or output

- Configuration of messages that are triggered with a condition
- Configuration of dynamics defaults and limits for all axes
- Configuration of software and hardware limits
- Activate or deactivate the axis, simulation mode or the motor holding brake
- Configure the kinematics type and the blocked zones
- Change the HMI runtime language or stop the runtime

Diagnostics

- Display status of all configured inputs, outputs, variables, and conditions
- Display the status of the program, zones, and the axes
- Display errors of the program, zones, and the axes and acknowledge them
- Show system diagnostics

1.3 Components used

This application example (TIA V17) has been created and tested with the following hard- and software components:

Table 1-1: Hardware and Software components

Component	Number	Article number	Note
SIMATIC Cartesian Portal V1.0	1	6ES7823-0GE01-1AA0	Runtime License via Online Software Delivery (OSD)
SIMATIC S7-1511T-1 PN	1	6ES7 511-1TK01-0AB0	Other S7-1500(T) CPU with minimum code-work memory of 225 kByte
SIMATIC S7, MEMORY CARD FOR S7-1X00 CPU	1	6ES7954-8LL03-0AA0	3,3 V FLASH, 256 MB
KTP900F Mobile Panel	1	6AV2125-2JB23-0AX0	Other Comfort or Mobile Panels
SIMATIC WinCC Advanced V17	1	6AV2102-0AA07-0AA5	SIMATIC WinCC Comfort V17
SIMATIC STEP 7 Professional V17 (TIA Portal)	1	6ES7822-1AA07-0YA5	

You can purchase these components from the Siemens Industry Mall.

NOTE

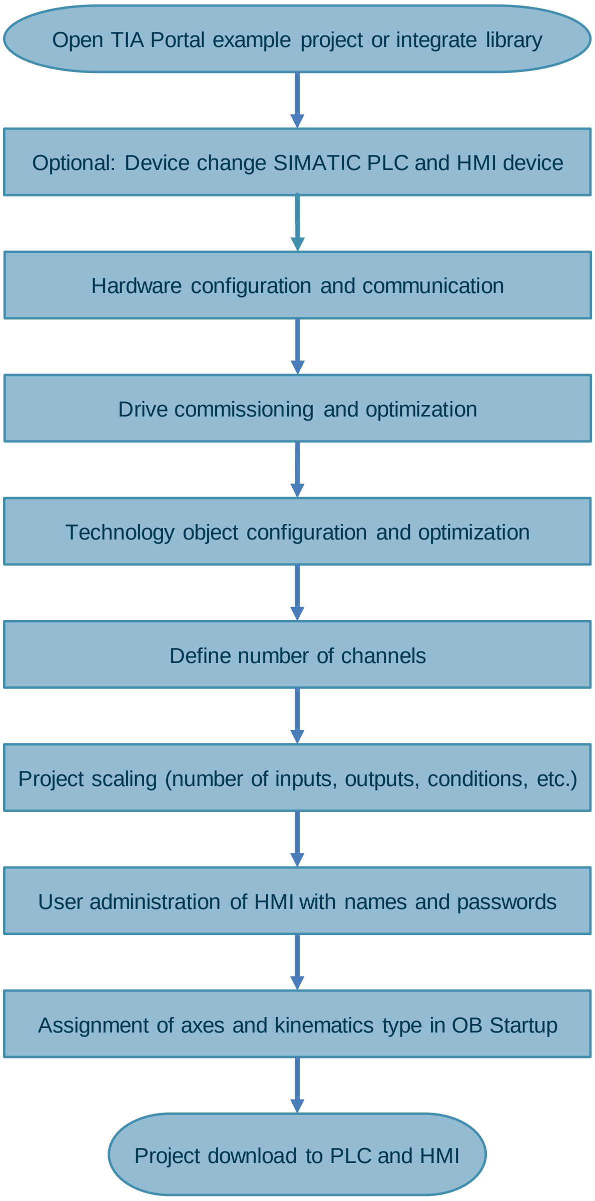
The application can run on SIMATIC CPU with FW 2.9 or higher. The minimum CPU is 1511T (V17, FW V2.9) or 1511 (V18, FW V3.0).

The application can run on SIMATIC Mobile Panels and on SIMATIC Comfort Panels with a minimum display size of 7 inch and V17 firmware.

2 Commissioning in TIA Portal

Before you can use the SIMATIC Cartesian Portal application on HMI, you must adapt the TIA Portal project to your machine. The following steps are mandatory to use the SIMATIC Cartesian Portal successfully.

Figure 2-1: Prerequisites SIMATIC Cartesian Portal in TIA Portal


Open TIA Portal example project or integrate library
Optional: Device change SIMATIC PLC and HMI device
Hardware configuration and communication
Drive commissioning and optimization
Technology object configuration and optimization
Define number of channels
Project scaling (number of inputs, outputs, conditions, etc.)
User administration of HMI with names and passwords
Assignment of axes and kinematics type in OB Startup
Project download to PLC and HMI

2.1 Hardware configuration of inputs and outputs

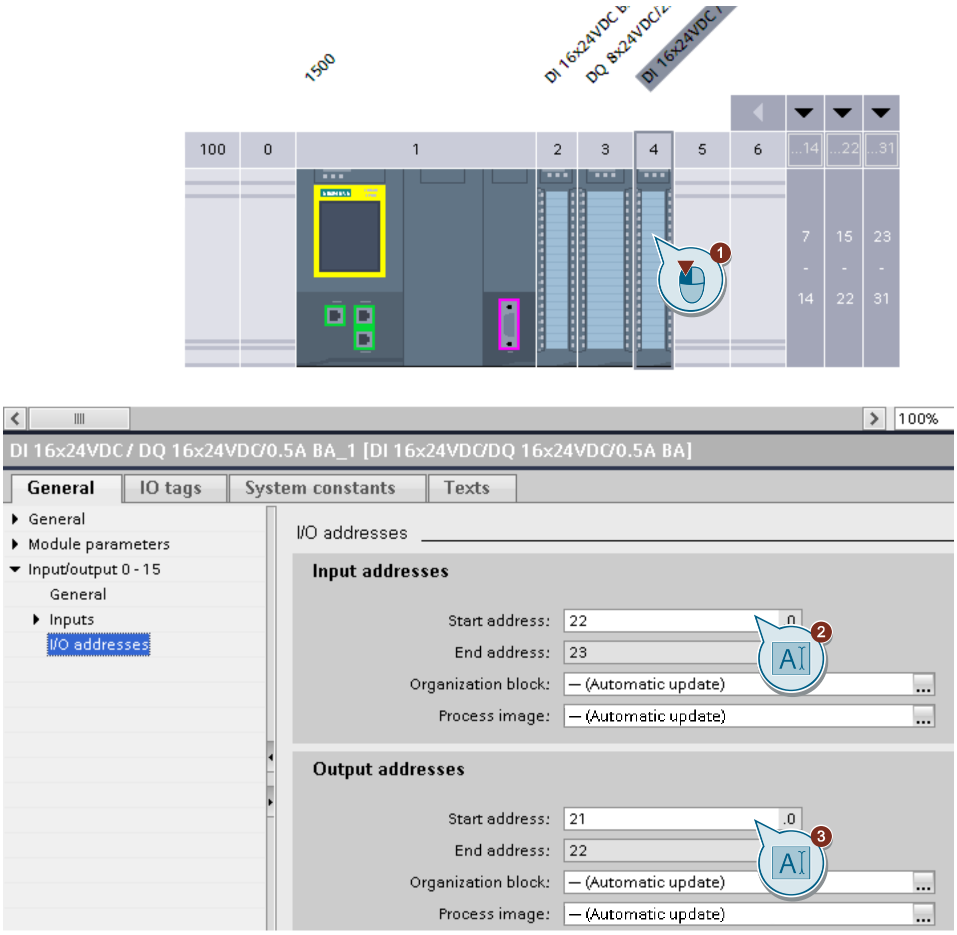
The addresses of all input and output modules must be configured in TIA Portal. If necessary, you must define IO tags for the hardware limit switches and the external zero marks for active homing. All other tags for inputs and outputs can be defined directly on the HMI.

The input and output function test and naming can be done via the faceplate ‘Configuration’ on the HMI (cf. chapter 3.7). You must use the configured address space from the TIA Portal project on HMI.

To set the address space of a module

7. Select the DI or DQ module.
8. Type in a valid start address for digital inputs.
9. Type in a valid start address for digital outputs.
10. Define the IO tags for hardware limit switches and the external zero marks.

Figure 2-2: Set input and output start addresses



DI 16x24VDC / DQ 16x24VDC/0.5A BA_1 [DI 16x24VDC/DQ 16x24VDC/0.5A BA]

General | IO tags | System constants | Texts

- General
- Module parameters
- Input/output 0 - 15
  - General
  - Inputs
  - I/O addresses

I/O addresses

Input addresses

Start address: 22 .0
End address: 23
Organization block: --- (Automatic update)
Process image: --- (Automatic update)

Output addresses

Start address: 21 .0
End address: 22
Organization block: --- (Automatic update)
Process image: --- (Automatic update)

2.2 Drive commissioning and optimization

The SIMATIC Cartesian Portal application is independent from the drive type used in the machine. You must parametrize and optimize the drives. You can use optimizing functions, e.g., one-button-tuning.

The solution works with the following Siemens drives

- SINAMICS V90
- SINAMICS S210
- SINAMICS S120

Third party drives must meet the following requirements

- PROFINET interface
- PROFIdrive Application Class 4 (AC4) (e.g. PROFIdrive-Telegram 3 or 5)
- Isochronous mode and isochronous real time (IRT) are recommended

2.3 Technology objects

2.3.1 Technology object configuration

Supported technology objects

The SIMATIC Cartesian Portal enables the programming of the following technology objects on the HMI.

- TO_PositioningAxis

Technology object axis configuration

⚠ CAUTION

The axes configurations in the delivered TIA project are configured for one specific kinematics and drive parameterization. This configuration cannot be used by default for other mechanics and other drives. You must adapt the configuration to your kinematics.

Please optimize and test your axes before using the SIMATIC Cartesian Portal solution to avoid uncontrolled movement of the axes.

It is recommended to reduce the following error to a minimum and to avoid oscillations of the axes.

SIMATIC Cartesian Portal supports the change of selected configuration parameters of the technology objects directly on the HMI.

Table 2-1: TO axis configuration

Configuration	TIA Portal	HMI
Drive/encoder connection	Mandatory	No configuration
Mechanics	Mandatory	No configuration
Leading value interconnections	Mandatory	No configuration
Active homing sequence	Mandatory	No configuration
Control loop	Mandatory	No configuration
Hardware limit switch tags	Mandatory	No configuration
Position monitoring	Mandatory	No configuration
Following error	Mandatory	No configuration
Standstill signal	Mandatory	No configuration
Hardware limit switch enable	Offline configuration	Configuration at runtime
Software limit switches	Offline configuration	Configuration at runtime
Simulation	Offline configuration	Configuration at runtime
Dynamic defaults/limits	Offline configuration	Configuration at runtime
Emergency stop	Offline configuration	Configuration at runtime

All technology objects should be tested and optimized with the control panel in TIA Portal before the SIMATIC Cartesian Portal is used.

Number of axes

The SIMATIC Cartesian Portal supports up to six kinematics axes and up to 25 additional auxiliary axes per channel.

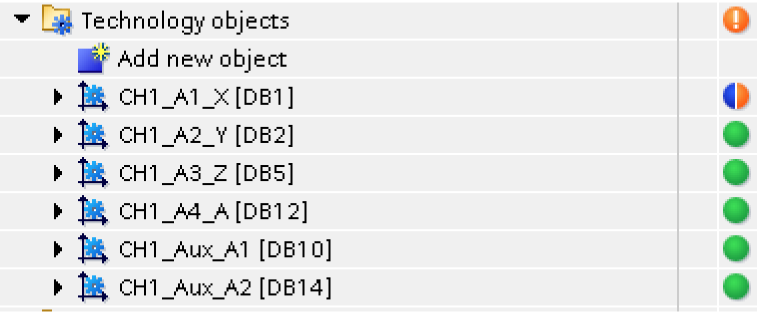
2.3.2 Keep technology object consistency

Configuration changes on the HMI are written into the load memory of the technology objects. The configuration changes remain after Power-Off/On of the CPU.

NOTICE	**The configuration changes on the HMI are not synchronized with the offline TIA project. The data between the TIA project and the load memory in the PLC are inconsistent after HMI configuration changes.** After a new download without synchronization of the technology objects from TIA Portal to the PLC the changed configurations will be lost. An upload of the technology objects must be executed before a download to the PLC can be executed to keep the data consistency. Optionally the manual data block synchronization of TIA Portal can be used.

The inconsistent technology objects are displayed in the projects tree.

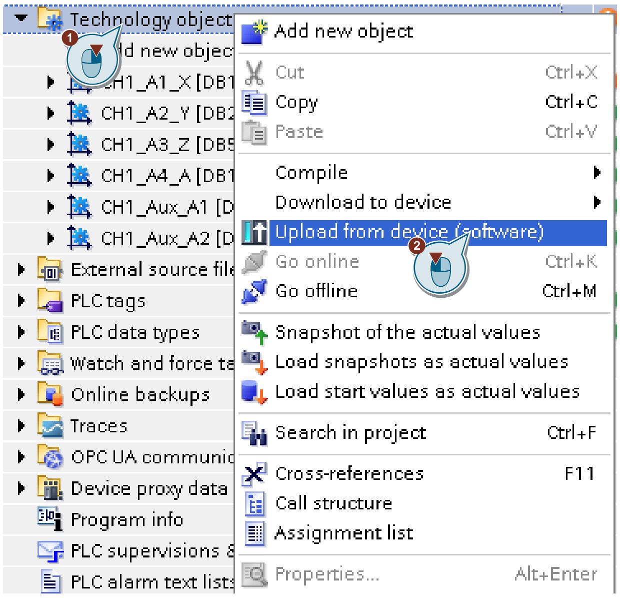
Figure 2-3: Inconsistent technology object CH1_A1_X



Upload technology objects

1. Go online on the PLC.
2. Click the right mouse button on the folder technology objects.
3. Click ‘Upload from device’.
   Afterwards the TIA project technology objects are consistent with the PLC project.

Figure 2-4: Upload technology objects from device

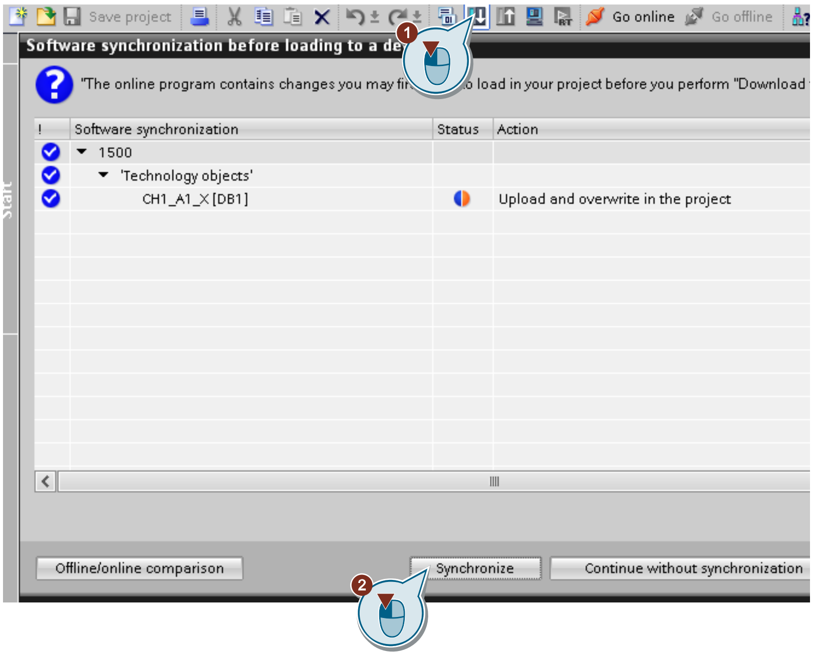


Synchronize technology objects manually

1. Download the project.
2. Click the synchronize button.
   Afterwards the TIA project technology objects are consistent with the PLC project.

NOTE Use the upload function if the synchronization is not offered before downloading the project.

Figure 2-5: Synchronize technology objects



Software synchronization before loading to a de

"The online program contains changes you may fir o load in your project before you perform "Download

!	Software synchronization	Status	Action
✓	1500		
✓	'Technology objects'		
✓	CH1_A1_X [DB1]	◐	Upload and overwrite in the project

2.4 Integration of the SIMATIC Cartesian Portal library

Prerequisites

The SIMATIC Cartesian Portal library has no dependencies to other libraries and works standalone. To get started

1. Open the global library folder.
2. Open the library ‘SIMATICCartesianPortal_V_x_y_z_Library_Vxy’ > Master copies.

2.4.1 Integration of the library in the PLC

The library LSCP contains a complete PLC in the ‘Master copies’ folder ‘_CompleteProject’ and single elements for the integration into an existing PLC.

Import of the complete PLC

To integrate a complete PLC

1. Navigate to the folder ‘_CompleteProject’ > ‘1500/1500T/DriveController’ or ‘OpenController’ > ‘PLC’ depending on the desired PLC type.
2. Copy the PLC ‘1511T’/’1511’ or ‘PC_System_1’ via Drag & Drop into the TIA project.
3. Compile the PLC.
4. Proceed with chapter 2.4.2.

Import into an existing PLC

The following steps are required to import the SIMATIC Cartesian Portal to an existing PLC project.

1. Navigate to the folder ‘PLC’.
2. Copy the folder ‘LSCP_Tags’ with Drag & Drop into the ‘PLC tags’ in the PLC.
3. Copy the folder ‘LSCP_Types’ in ‘LSCP_1500/1500T/DriveController’ or ‘LSCP_OpenController’ depending on the PLC type with Drag & Drop into the ‘PLC data types’ in the PLC.
4. Copy the folder ‘LSCP_Blocks’ in ‘LSCP_1500/1500T/DriveController’ or ‘LSCP_OpenController’ depending on the PLC type with Drag & Drop into the ‘Program blocks’ in the PLC.
5. Compile the PLC project.

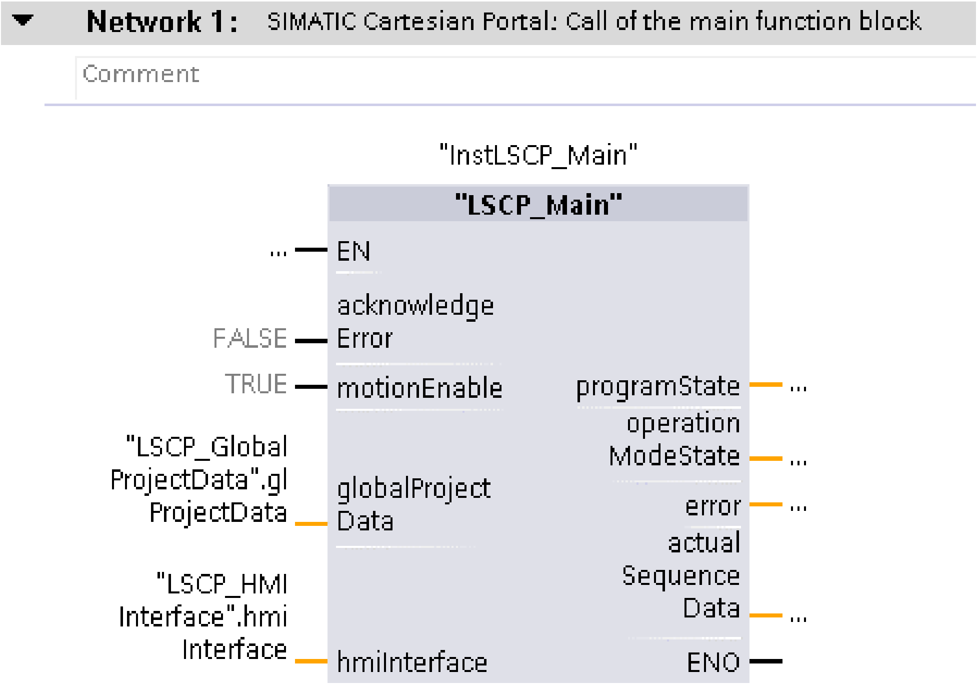
Call the SCP

6. Copy the Block ‘LSCP_MainOB’ from ‘LSCP_Blocks_Additional’ with Drag & Drop into the ‘Program blocks’ in the PLC
7. Proceed with chapter 2.4.2.

or

6. Call the FB ‘LSCP_Main’ with the single instance ‘instLSCP_Main’ in a ‘Program cycle’ OB like OB1.
7. Connect ‘”LSCP_GlobalProjectData”.glProjectData’ with the input ‘globalProjectData’.
8. Connect ‘”LSCP_HMIInterface”.hmiInterface’ to the input ‘hmiInterface’.
9. Proceed with chapter 2.4.2.

Figure 2-6: Call LSCP_Main FB in LSCP_MainOB



Network 1: SIMATIC Cartesian Portal: Call of the main function block

Comment

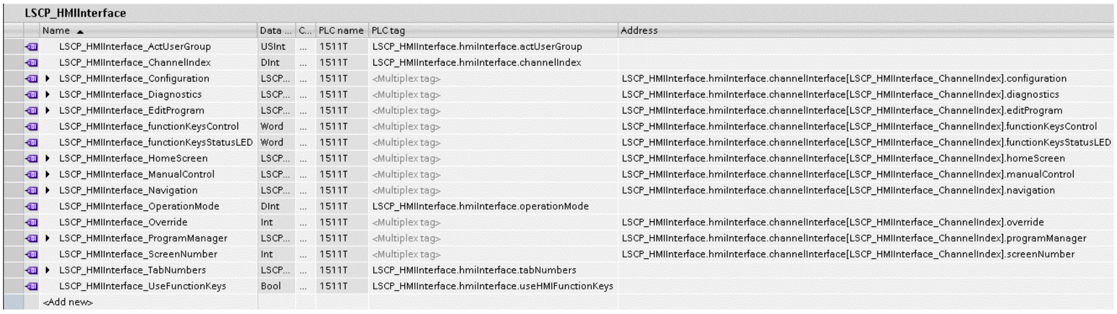
2.4.2 Integration of the library into the HMI

The library LSCP contains a complete panel in the ‘Master copies’ folder ‘_CompleteProject’ and single elements for the integration into an existing Panel.

Import of the complete Panel

1. Navigate to the folder ‘_CompleteProject’ > ‘1500/1500T/DriveController’ or ‘OpenController’ > ‘HMI’ depending on the used PLC type. This is only relevant if the PLC was used from ‘_CompleteProject’ since step 4 can be skipped.
2. Copy the panel ‘Comfort 9’’ 1511’ or ‘Comfort 9’’ OC’ via Drag & Drop into the TIA project.
3. Establish an HMI connection between the panel and the PLC.
4. Connect the HMI tags in the HMI tag table ‘LSCP_HMIInterface’ to the corresponding PLC tags like shown in Figure 2-7. The multiplex tags must be configured in two stages: First assign the HMI tag to the PLC tag of channel zero. Second multiplex the HMI tag by selecting the ‘LSCP_HMIInterface_ChannelIndex’ as the index tag (cf. Figure 2-8).
5. Compile the HMI.

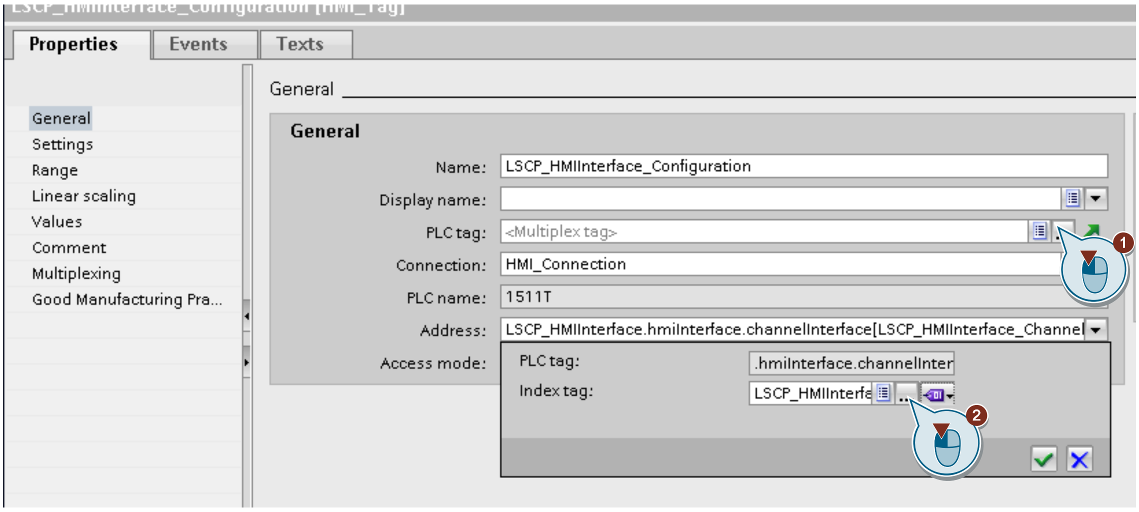
Figure 2-7: Connect LSCP HMI tags in ‘LSCP_HMIInterface’



LSCP_HMIInterface

Name ▲	Data ...	C...	PLC name	PLC tag	Address
LSCP_HMIInterface_ActUserGroup	USInt	...	1511T	LSCP_HMIInterface.hmiInterface.actUserGroup	
LSCP_HMIInterface_ChannelIndex	DInt	...	1511T	LSCP_HMIInterface.hmiInterface.channelIndex	
▸ LSCP_HMIInterface_Configuration	LSCP...	...	1511T	<Multiplex tag>	LSCP_HMIInterface.hmiInterface.channelInterface[LSCP_HMIInterface_ChannelIndex].configuration
▸ LSCP_HMIInterface_Diagnostics	LSCP...	...	1511T	<Multiplex tag>	LSCP_HMIInterface.hmiInterface.channelInterface[LSCP_HMIInterface_ChannelIndex].diagnostics
▸ LSCP_HMIInterface_EditProgram	LSCP...	...	1511T	<Multiplex tag>	LSCP_HMIInterface.hmiInterface.channelInterface[LSCP_HMIInterface_ChannelIndex].editProgram
LSCP_HMIInterface_functionKeysControl	Word	...	1511T	<Multiplex tag>	LSCP_HMIInterface.hmiInterface.channelInterface[LSCP_HMIInterface_ChannelIndex].functionKeysControl
LSCP_HMIInterface_functionKeysStatusLED	Word	...	1511T	<Multiplex tag>	LSCP_HMIInterface.hmiInterface.channelInterface[LSCP_HMIInterface_ChannelIndex].functionKeysStatusLED
▸ LSCP_HMIInterface_HomeScreen	LSCP...	...	1511T	<Multiplex tag>	LSCP_HMIInterface.hmiInterface.channelInterface[LSCP_HMIInterface_ChannelIndex].homeScreen
▸ LSCP_HMIInterface_ManualControl	LSCP...	...	1511T	<Multiplex tag>	LSCP_HMIInterface.hmiInterface.channelInterface[LSCP_HMIInterface_ChannelIndex].manualControl
▸ LSCP_HMIInterface_Navigation	LSCP...	...	1511T	<Multiplex tag>	LSCP_HMIInterface.hmiInterface.channelInterface[LSCP_HMIInterface_ChannelIndex].navigation
LSCP_HMIInterface_OperationMode	DInt	...	1511T	LSCP_HMIInterface.hmiInterface.operationMode	
LSCP_HMIInterface_Override	Int	...	1511T	<Multiplex tag>	LSCP_HMIInterface.hmiInterface.channelInterface[LSCP_HMIInterface_ChannelIndex].override
▸ LSCP_HMIInterface_ProgramManager	LSCP...	...	1511T	<Multiplex tag>	LSCP_HMIInterface.hmiInterface.channelInterface[LSCP_HMIInterface_ChannelIndex].programManager
LSCP_HMIInterface_ScreenNumber	Int	...	1511T	<Multiplex tag>	LSCP_HMIInterface.hmiInterface.channelInterface[LSCP_HMIInterface_ChannelIndex].screenNumber
▸ LSCP_HMIInterface_TabNumbers	LSCP...	...	1511T	LSCP_HMIInterface.hmiInterface.tabNumbers	
LSCP_HMIInterface_UseFunctionKeys	Bool	...	1511T	LSCP_HMIInterface.hmiInterface.useHMIFunctionKeys	
<Add new>					

Figure 2-8: Connect multiplex variables



General

General

Name: LSCP_HMIInterface_Configuration

Display name:

PLC tag: <Multiplex tag>

Connection: HMI_Connection

PLC name: 1511T

Address: LSCP_HMIInterface.hmiInterface.channelInterface[LSCP_HMIInterface_Channel

Access mode:

PLC tag: .hmiInterface.channelInter

Index tag: LSCP_HMIInterfa

Import single HMI elements to an existing panel

1. Establish an HMI connection between the Panel and the PLC.
2. Navigate to the folder ‘HMI’.
3. Copy the HMI tags from the folder ‘HMI tags’ to the HMI tags folder.
4. Connect the HMI tags in the ‘LSCP_HMIInterface’ tag table to the corresponding PLC tags (cf. Figure 2-7 and Figure 2-8).
5. Copy the scripts from the folder ‘Scripts ’to the folder ‘Scripts’ in the HMI

6. Copy the screen templates from the folder ‘Screen management’ > ‘Templates’ to the folder ‘Screen management’ > ‘Templates’.
7. Copy the pop-up screens from the folder ‘HMI’ > ‘Screen management’ > ‘Pop-up screens’ to the folder ‘Screen management’ > ‘Pop-up screens’.
8. Copy the SCP screens in the ‘Screens’ folder to the folder ‘Screens’ in the HMI.
9. Copy scheduled tasks in the folder ‘Scheduled tasks’ into the HMI scheduled tasks.
10. (Optional) Copy user administration in the folder ‘User administration’. Start with ‘Groups’ and then ‘Users’.
11. Compile the HMI.

2.5 Project scaling

Project size

The following table shows the required memory size of the PLC project for different PLCs. The table shows an exemplary configuration for each PLC with one kinematics (channel). If more than one kinematics (channel) is used, the available data per channel must be divided through the number of channels.

Table 2-2: Example configuration and resulting project size for one channel

	1511-T	1512C	OC 1515 SP	1515-T	1516-T	1517-T	1504D TF	1507D TF
Slots	10	10	40	40	40	50	50	50
Commands	50	50	80	90	90	110	110	110
Outputs	50	50	80	80	80	110	110	110
Inputs	50	50	80	80	80	110	110	110
Conditions	50	50	80	80	80	110	110	110
Variables	50	50	80	80	80	110	110	110
Labels	50	50	80	80	80	110	110	110
Points	50	50	80	80	80	110	110	110
Zones	10	10	10	10	10	10	10	10
Code work memory	97%	87%	16%	29%	14%	7%	11%	4%
Data work memory	42%	42%	21%	37%	22%	19%	38%	8%
Retain memory	97%	97%	97%	93%	93%	95%	95%	95%

Scaling of the project with global constants

If you want to raise or reduce the data of the SIMATIC Cartesian Portal you must adapt the global constant values.

The constants can be adapted to raise the number of commands, variables, points, and other configuration parameters. The following table displays the possible scaling options for the SCP. The adaptions are effective for all channels.

The project tags of the SCP are defined in the tag table PLC tags > ‘LSCP_Tags’ > ‘LSCP_ProjectConfiguration’. You can adapt these constants to the required values of your application. ‘MAX’ means, that the constant can be theoretically extended to the maximum value of the data type. Practically the retain memory of the PLC limits the number of elements and the maximum values should always be configured to the needs of the application.

Table 2-3: Project scaling of the SCP in the tag table ‘LSCP_ProjectConfiguration’

Constant name	Description	Default	Minimum	Maximum
LSCP_NO_OF_OUTPUTS	Number of Outputs - 1.	39	7	MAX
LSCP_NO_OF_INPUTS	Number of Inputs – 1.	39	7	MAX
LSCP_NO_OF_CHANNELS	Number of channels + 1.	0	0	MAX
LSCP_NO_OF_ PROGRAM_SLOTS	Number of program slots – 1.	4	1	MAX
LSCP_NO_OF _COMMANDS	Number of commands - 1.	49	15	MAX
LSCP_NO_OF_LABELS	Number of labels - 1.	49	1	MAX
LSCP_NO_OF_POINTS_ AT_POINT_TABLE	Number of points at point table.	49	1	MAX
LSCP_NO_OF_ CONDITIONS	Number of conditions + 6.	39	7	MAX
LSCP_NO_OF_ VARIABLES	Number of variables – 1.	39	Number of connected Kinematics axes + No of connected aux axes + 7..x.	MAX
LSCP_NO_OF_ZONES	Number of zones – 1.	9	0	MAX
LSCP_NO_OF_NAV_LEVELS	0: One navigation level, no user tabs, 1: Two navigation levels, 6 additional user tabs, 2: Three navigation levels, 12 additional user tabs.	0	0	2
LSCP_NO_OF_HOMESCREEN_BUTTONS	Number of visible home screen buttons.	1	0	9
LSCP_USE_HMI_FUNCTION_KEYS	Use the function keys of the HMI to control the program execution.	TRUE	-	-
LSCP_MOTION_ENABLE_FULL_OP_MODE	TRUE: Motion enable is evaluated in MCtrl in Full Operation mode, FALSE: Motion enable is not evaluated in MCtrl in Full Operation Mode	TRUE	-	-

Set number of kinematics and channels

The number of kinematics is directly related to the number of channels. There can only be one kinematics per channel. To increase the number of kinematics the global constant ‘LSCP_NO_OF_CHANNELS’ in ‘PLC tags’ > ‘LSCP_Tags’ > ‘LSCP_ProjectConfiguration’ must be increased.

NOTE

It is also possible to have no kinematics in a channel. There must be at least one used axis per channel otherwise the SCP will not start.

2.6 User administration of the HMI

User login and passwords

The SIMATIC Cartesian Portal offers four default user roles (cf. Table 2-4). You can change the usernames and passwords in TIA Portal in the user administration of the HMI and on the panel. The user level is 0, if no user is logged in.

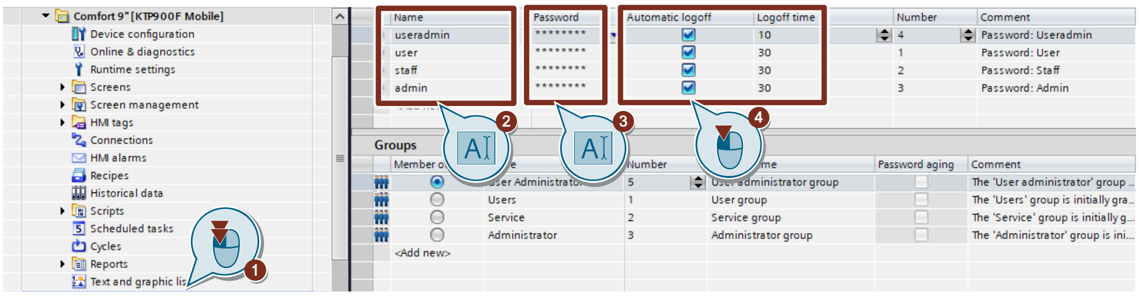
Table 2-4: Default user roles SIMATIC Cartesian Portal

Name	Password	User level	Automatic log off
useradmin	Useradmin	5	10 min
admin	Admin	3	30 min
staff	Staff	2	30 min
user	User	1	30 min

To change the user administration in the TIA Portal (cf. Figure 2-9):

1. Open the user administration of the panel.
2. Adapt the usernames.
3. Adapt the passwords.
4. Change the automatic logoff time or disable the automatic logoff function.

Figure 2-9: User administration of the HMI in the TIA Portal



Name	Password	Automatic logoff	Logoff time	Number	Comment
useradmin	********	☑	10	4	Password: Useradmin
user	********	☑	30	1	Password: User
staff	********	☑	30	2	Password: Staff
admin	********	☑	30	3	Password: Admin

Member o	e	Number	me	Password aging	Comment
●	User Administrator	5	User administrator group	□	The 'User administrator' group ...
○	Users	1	User group	□	The 'Users' group is initially gra...
○	Service	2	Service group	□	The 'Service' group is initially g...
○	Administrator	3	Administrator group	□	The 'Administrator' group is ini...
<Add new>					

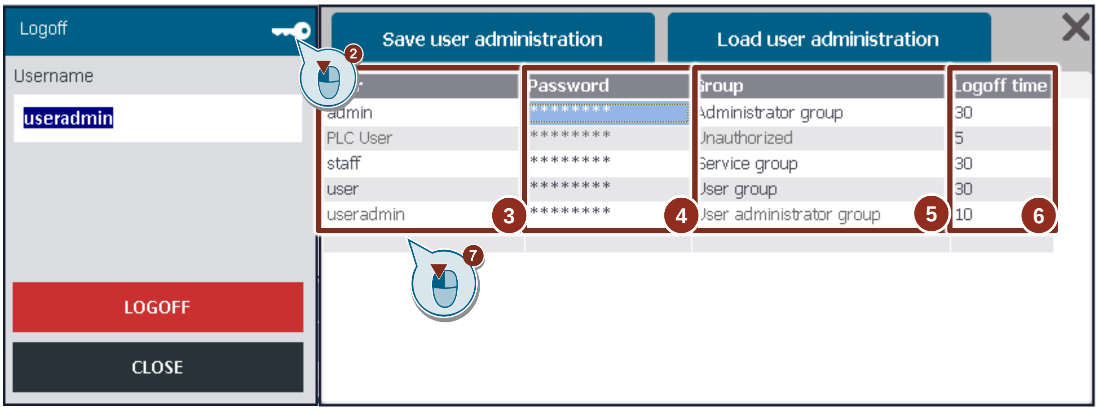
To change the user administration on the HMI (cf. Figure 2-10):

1. Log in as ‘useradmin’
2. Reopen the log in screen and click the key icon
3. Adapt the usernames
4. Adapt the passwords
5. Change the groups
6. Adapt the logoff time
7. Create a new user

NOTE

In the user administration on the HMI, you can also reassign a group level to a user if it was lost due to exceeding the limit of logon attempts.

Figure 2-10: User administration on the HMI



Logoff

Username

useradmin

Save user administration

Load user administration

r	Password	roup	Logoff time
admin	********	dministrator group	30
PLC User	********	nauthorized	5
staff	********	ervice group	30
user	********	ser group	30
useradmin	********	ser administrator group	10

LOGOFF

CLOSE

Required user levels

The SIMATIC Cartesian Portal has implemented a user administration with different user levels. The group level of the logged in user determines which parts of the SCP can be used.

Table 2-5 shows the required default user levels from the SIMATIC Kinematics Integrator. The ‘X’ indicates that the user can utilize the function on the HMI. These default values can be changed by changing the value of the constant in the ‘LSCP_UserLevel’ tag table in the ‘LSCP_Tags’ in the PLC.

Table 2-5: User levels in the tag table ‘LSCP_UserLevel’

Constant name LSCP_USERLEVEL_	Function	useradmin (5)	admin (3)	staff (2)	user (1)	No login (0)
VARIABLE_DEFINITION	Access the variable and message tab in the configuration	x	x	x		
CONFIGURATION	Access the axis and kinematics tab in the configuration	x	x			
MANUAL_CONTROL	Jog single axes in the manual control tab	x	x	x	x	
HOME_AXES	Home axes in the manual control tab	x	x			
CONFIGURE_POINTS	Configure points in the manual control tab	x	x	x		
TEACH_APPROACH_POINTS	Teach or approach points in the manual control tab	x	x			
EDIT_PROGRAM	Access the edit program tab	x	x	x		
PROGRAM_MANAGER	Access the program manager tab	x	x	x		
ACKNOWLEDGE_ERRORS	Acknowledge any kind of error	x	x	x		
STOP_RUNTIME	Stop the HMI runtime	x	x			

2.7 Adapt OB1 maximum cycle time

The maximum cycle time of OB1 is a configurable upper limit of the cyclic program runtime. In the SIMATIC Cartesian Portal some processes need a long execution time due to initialization processes.

NOTICE

To avoid a CPU-STOP the maximum time must be increased to a higher value.

For the SIMATIC S7-1511T CPU a minimum setting for the maximum cycle time of 500 ms is required. If you adapt the project size (see chapter 2.5), higher times can be required.

2.8 Assignment of technology objects in the startup OB

The technology objects must be assigned to the corresponding channels in a startup OB. Their assignment cannot be changed from the HMI during runtime.

NOTE

The startup OB ‘LSCP_Startup[OB127]’ is a template provided by the application example and with the library under ‘PLC’ > ‘LSCP_Blocks_Additional’ and can be used directly to assign the technology objects. Every other or an already existing startup OB can be used as well.

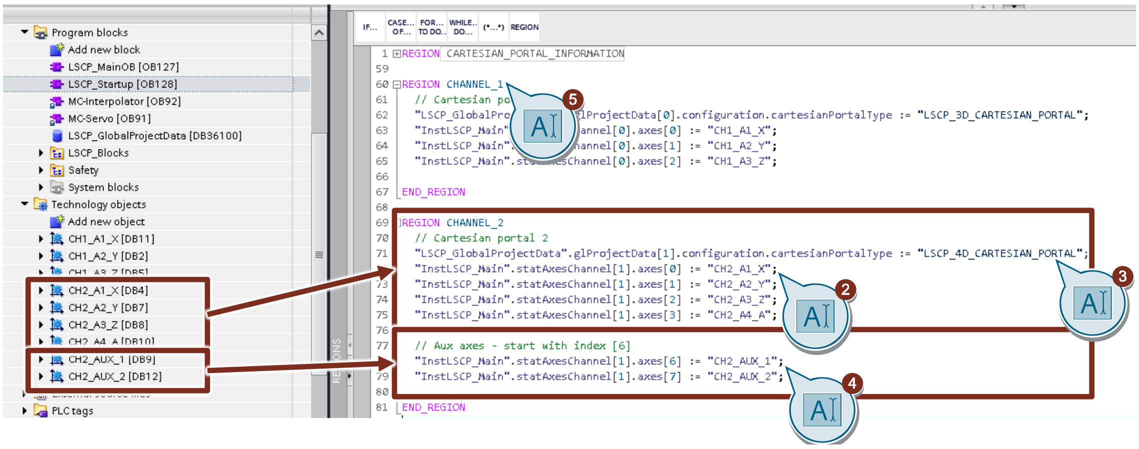
To assign the technology object (cf. Figure 2-11):

1. Open the startup OB.
2. Assign the kinematics axes to the variable ‘InstLSCP_Main.statAxesChannel[channelIndex].axes[0 – 5]’.
3. Assign the kinematics type to the variable ‘LSCP_GlobalProjectData.glProjectData[channelIndex].configuration.kinematicsType’. A description of the available kinematics types can be found in the region ‘KINEMATICS_INFORMATION’ in the ‘LSCP_Startup’. The kinematics type can be changed on the HMI during runtime.
4. Assign the auxiliary axes to the array ‘InstLSCP_Main.statAxesChannel[channelIndex].axes[6 - x]’.
5. Repeat steps 2, 3, and 4 for each channel. Use the corresponding channel number as the ‘channelIndex’.

NOTE

For an optimal experience on the HMI leave no gaps in the array when assigning the kinematics axes (starting from 0) and the auxiliary axes (starting from 6).

Figure 2-11: Assign technology objects in ‘LSCP_Startup’



```
REGION CARTESIAN_PORTAL_INFORMATION

REGION CHANNEL_1
  // Cartesian po
  "LSCP_GlobalPro                 lProjectData[0].configuration.cartesianPortalType := "LSCP_3D_CARTESIAN_PORTAL";
  "InstLSCP_Main"                 annel[0].axes[0] := "CH1_A1_X";
  "InstLSCP_Main".                hannel[0].axes[1] := "CH1_A2_Y";
  "InstLSCP_Main".st        esChannel[0].axes[2] := "CH1_A3_Z";

END_REGION

REGION CHANNEL_2
  // Cartesian portal 2
  "LSCP_GlobalProjectData".glProjectData[1].configuration.cartesianPortalType := "LSCP_4D_CARTESIAN_PORTAL";
  "InstLSCP_Main".statAxesChannel[1].axes[0] := "CH2_A1_X";
  "InstLSCP_Main".statAxesChannel[1].axes[1] := "CH2_A2_Y";
  "InstLSCP_Main".statAxesChannel[1].axes[2] := "CH2_A3_Z";
  "InstLSCP_Main".statAxesChannel[1].axes[3] := "CH2_A4_A";

  // Aux axes - start with index [6]
  "InstLSCP_Main".statAxesChannel[1].axes[6] := "CH2_AUX_1";
  "InstLSCP_Main".statAxesChannel[1].axes[7] := "CH2_AUX_2";

END_REGION
```

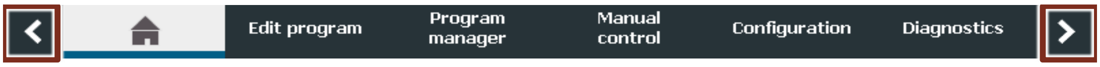
2.9 HMI navigation settings and extension

The HMI navigation faceplate offers various settings to customize its look and size.

Extension of the navigation

By default, the navigation offers six tabs. These can be extended to 12 or 18 tabs by setting the global constant ‘LSCP_NO_OF_NAV_LEVELS’ in the tag table ‘LSCP_ProjectConfiguration’ to one or two respectively. When multiple navigation levels are available arrows on the left and right of the navigation bar appear that cycle through the navigation levels (cf. Figure 2-12).

Figure 2-12: Navigation level change on HMI



Connect a user screen to a tab and modify the tab order

Each tab calls a screen by addressing its unique number when clicked. The number of a screen can be modified in the screen properties (cf. Figure 2-13).

To connect a user screen to a tab:

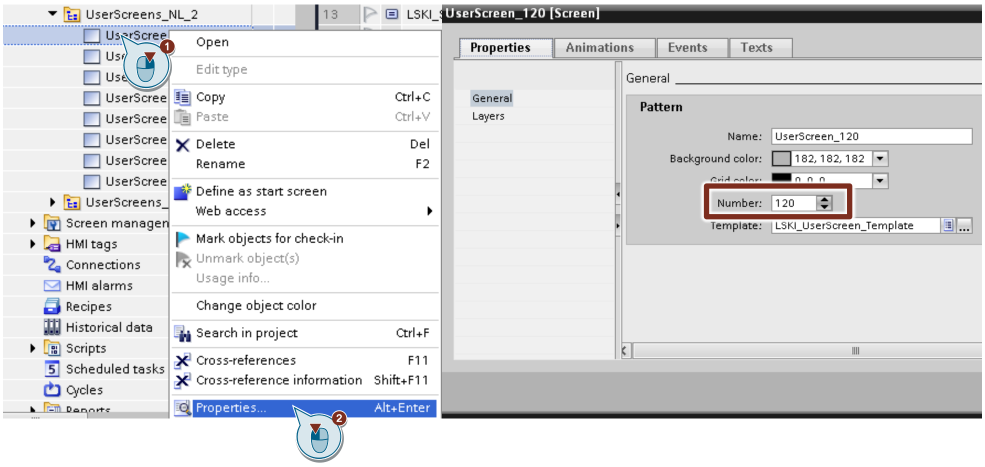
1. The screen number must be assigned to one of the user screen global constants in the tag table ‘LSCP_ScreenNumbers’.
2. This constant must then be assigned to a tab in the ‘LSCP_HMIInterface.hmiInterface.tabNumbers’ DB to connect it to this tab.

The predefined user screens in the example project are already connected to the corresponding tabs.

NOTE

> A screen like e.g., the ‘Diagnostics screen’ can be called on multiple navigation levels by assigning its constant to multiple tabs in the ‘LSCP_HMIInterface.hmiInterface.tabNumbers’.

Figure 2-13: Screen number of each screen



General

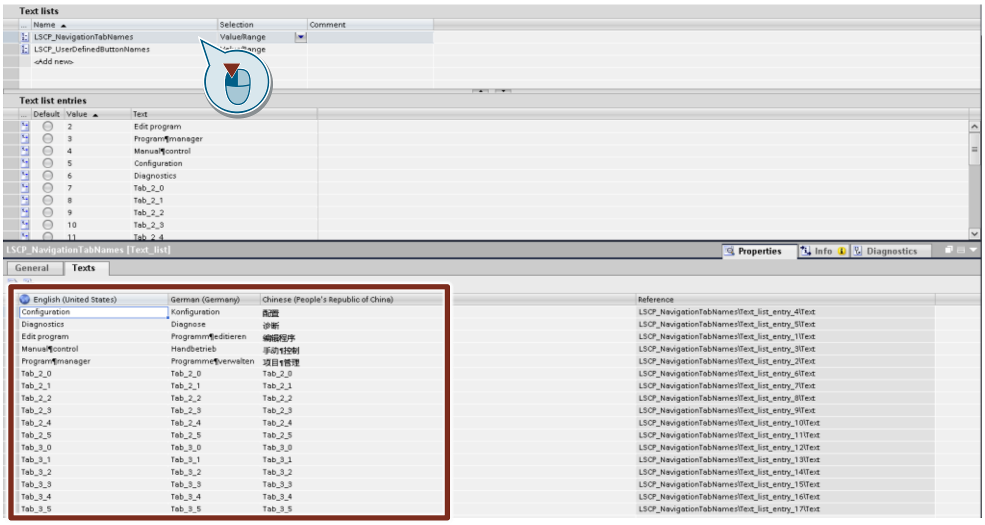
Name: UserScreen_120
Background color: 182, 182, 182
Grid color: 0, 0, 0
Number: 120
Template: LSKI_UserScreen_Template

NOTE After changing a screen number global constant, a recompile of the PLC with the option ‘Software (rebuild all)’ is necessary.

Edit the tab names

The name for each tab except the home screen (icon) can be configured in the HMI text list ‘LSCP_NavigationTabNames’ which can be found under ‘Text and graphics lists’ in the HMI menu. The values in the text list refer to the tab number starting from one and must not be changed. The tab names can be configured language dependent for each language that is enabled for the HMI. To see and edit the tab names in all activated languages, refer to Figure 2-14.

Figure 2-14: Edit the navigation tab names



Text lists

Name	Selection	Comment
LSCP_NavigationTabNames	Value/Range	
LSCP_UserDefinedButtonNames	Value/Range	
<Add new>		

Text list entries

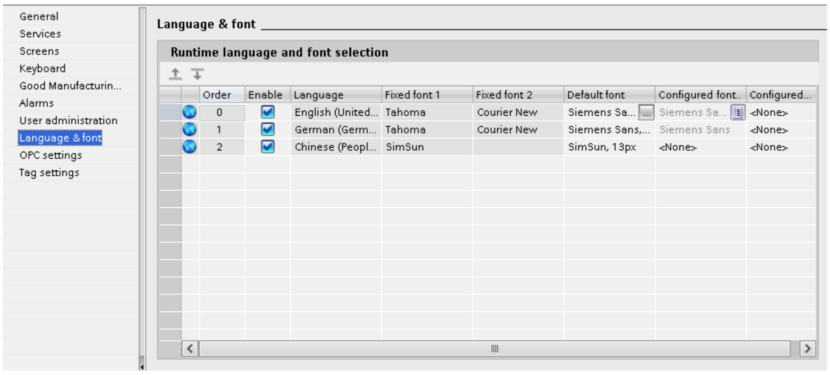
Default	Value	Text
○	2	Edit program
○	3	Program¶manager
○	4	Manual¶control
○	5	Configuration
○	6	Diagnostics
○	7	Tab_2_0
○	8	Tab_2_1
○	9	Tab_2_2
○	10	Tab_2_3
○	11	Tab_2_4

English (United States)	German (Germany)	Chinese (People's Republic of China)	Reference
Configuration	Konfiguration	配置	LSCP_NavigationTabNames\Text_list_entry_4\Text
Diagnostics	Diagnose	诊断	LSCP_NavigationTabNames\Text_list_entry_5\Text
Edit program	Programm¶editieren	编辑程序	LSCP_NavigationTabNames\Text_list_entry_1\Text
Manual¶control	Handbetrieb	手动¶控制	LSCP_NavigationTabNames\Text_list_entry_3\Text
Program¶manager	Programme¶verwalten	项目¶管理	LSCP_NavigationTabNames\Text_list_entry_2\Text
Tab_2_0	Tab_2_0	Tab_2_0	LSCP_NavigationTabNames\Text_list_entry_6\Text
Tab_2_1	Tab_2_1	Tab_2_1	LSCP_NavigationTabNames\Text_list_entry_7\Text
Tab_2_2	Tab_2_2	Tab_2_2	LSCP_NavigationTabNames\Text_list_entry_8\Text
Tab_2_3	Tab_2_3	Tab_2_3	LSCP_NavigationTabNames\Text_list_entry_9\Text
Tab_2_4	Tab_2_4	Tab_2_4	LSCP_NavigationTabNames\Text_list_entry_10\Text
Tab_2_5	Tab_2_5	Tab_2_5	LSCP_NavigationTabNames\Text_list_entry_11\Text
Tab_3_0	Tab_3_0	Tab_3_0	LSCP_NavigationTabNames\Text_list_entry_12\Text
Tab_3_1	Tab_3_1	Tab_3_1	LSCP_NavigationTabNames\Text_list_entry_13\Text
Tab_3_2	Tab_3_2	Tab_3_2	LSCP_NavigationTabNames\Text_list_entry_14\Text
Tab_3_3	Tab_3_3	Tab_3_3	LSCP_NavigationTabNames\Text_list_entry_15\Text
Tab_3_4	Tab_3_4	Tab_3_4	LSCP_NavigationTabNames\Text_list_entry_16\Text
Tab_3_5	Tab_3_5	Tab_3_5	LSCP_NavigationTabNames\Text_list_entry_17\Text

2.10 HMI languages

The SCP supports the HMI languages English, German, and Chinese. These are preconfigured in the application example and HMI copy template in the library (cf. Figure 2-15).

Figure 2-15: HMI Runtime settings – Language & font



Language & font

Runtime language and font selection

Order	Enable	Language	Fixed font 1	Fixed font 2	Default font	Configured font..	Configured...
0	☑	English (United...	Tahoma	Courier New	Siemens Sa...	Siemens Sa...	<None>
1	☑	German (Germ...	Tahoma	Courier New	Siemens Sans,...	Siemens Sans	<None>
2	☑	Chinese (Peopl...	SimSun		SimSun, 13px	<None>	<None>

NOTE	The order of the languages in Figure 2-15 matter for the selection on the HMI (cf. chapter 3.2.1). They must be in the displayed order in the runtime settings of the HMI. Another language can be added by extending the language selection in the 'Configuration' faceplate and adding the language to the HMI runtime settings.

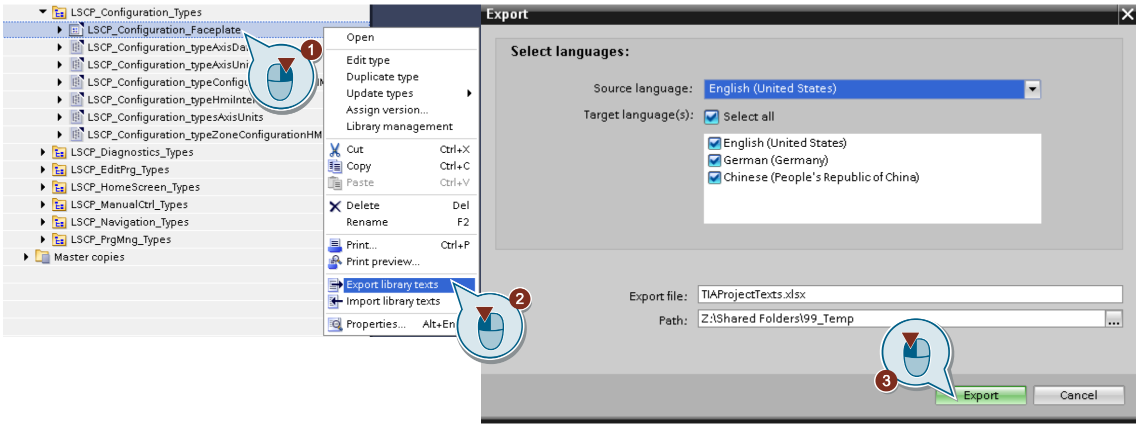
Export and import of HMI texts

To add another language the text lists of each faceplate and of the HMI in general must be adapted. It is best practice to follow these steps:

1. Add the new language to the 'Project languages' under 'Languages & resources'.
2. Enable the language in the 'Runtime settings' of the HMI under 'Language & font'.
3. Export the text list of each faceplate individually (cf. Figure 2-16).
4. Export the text list of the HMI in general (cf. Figure 2-17).
5. Translate the exported text list.
6. Import the translated text list of each faceplate (cf. Figure 2-18).
7. Import the translated text list of the HMI in general (cf. Figure 2-19).
8. Extend the 'Configuration' faceplate to the new language selection.

NOTICE	**Do not change anything on the HMI or a faceplate between the export and import of the text lists** If changes are made the import can fail and the text list must be exported and translated again.

Figure 2-16: Export text list of a faceplate



Select languages:

Source language: English (United States)

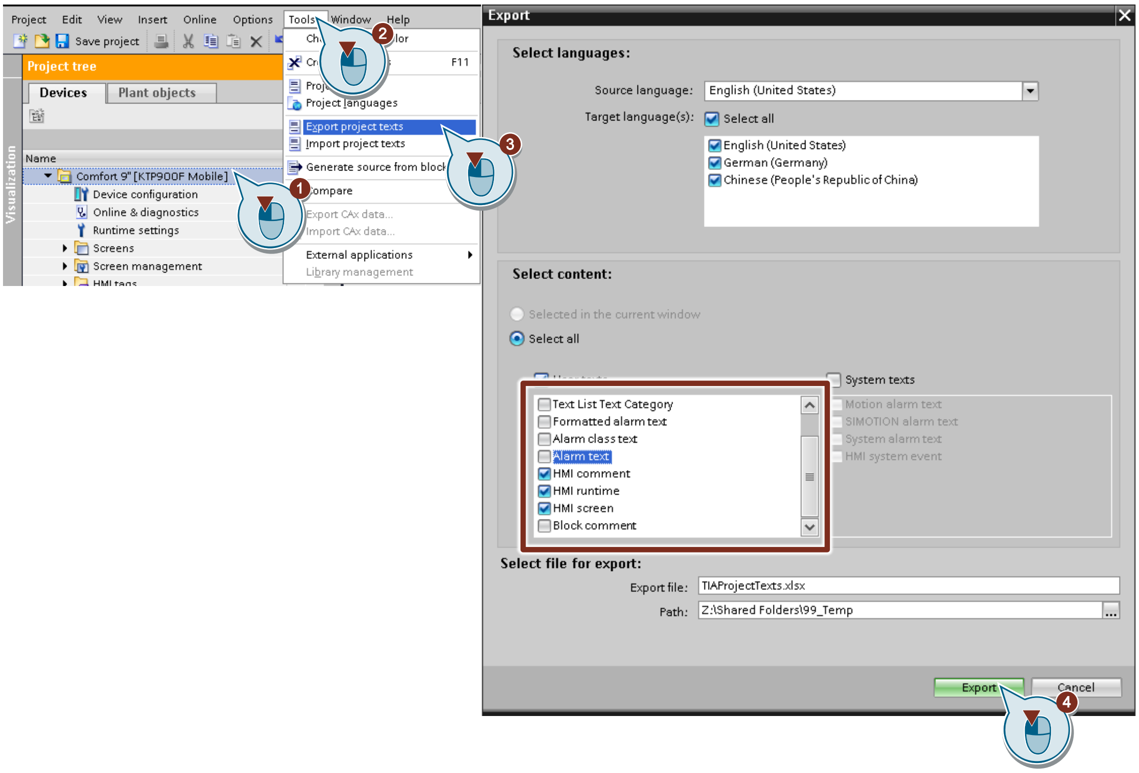
Target language(s): ☑ Select all

☑ English (United States)
☑ German (Germany)
☑ Chinese (People's Republic of China)

Export file: TIAProjectTexts.xlsx

Path: Z:\Shared Folders\99_Temp

Figure 2-17: Export general HMI text list



Export

Select languages:

Source language: English (United States)

Target language(s): ☑ Select all

☑ English (United States)
☑ German (Germany)
☑ Chinese (People's Republic of China)

Select content:

○ Selected in the current window
● Select all

□ Text List Text Category
□ Formatted alarm text
□ Alarm class text
□ Alarm text
☑ HMI comment
☑ HMI runtime
☑ HMI screen
□ Block comment

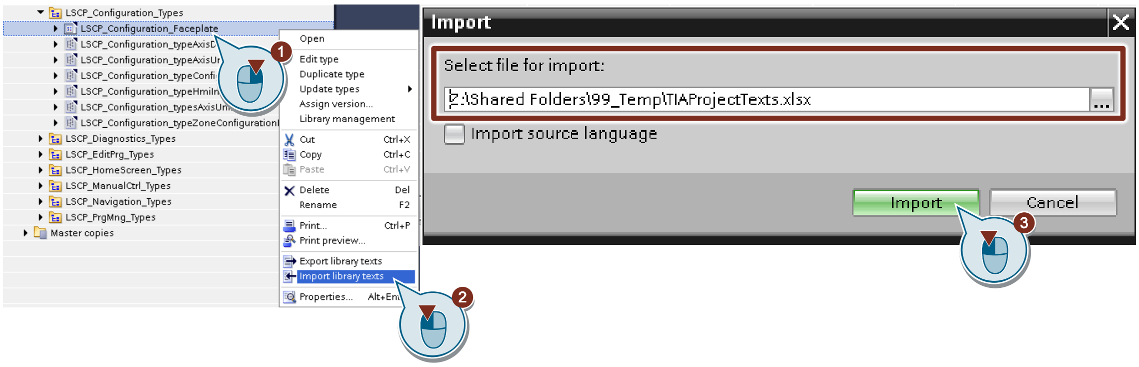
□ System texts
□ Motion alarm text
□ SIMOTION alarm text
□ System alarm text
□ HMI system event

Select file for export:

Export file: TIAProjectTexts.xlsx

Path: Z:\Shared Folders\99_Temp

Figure 2-18: Import text list of a faceplate



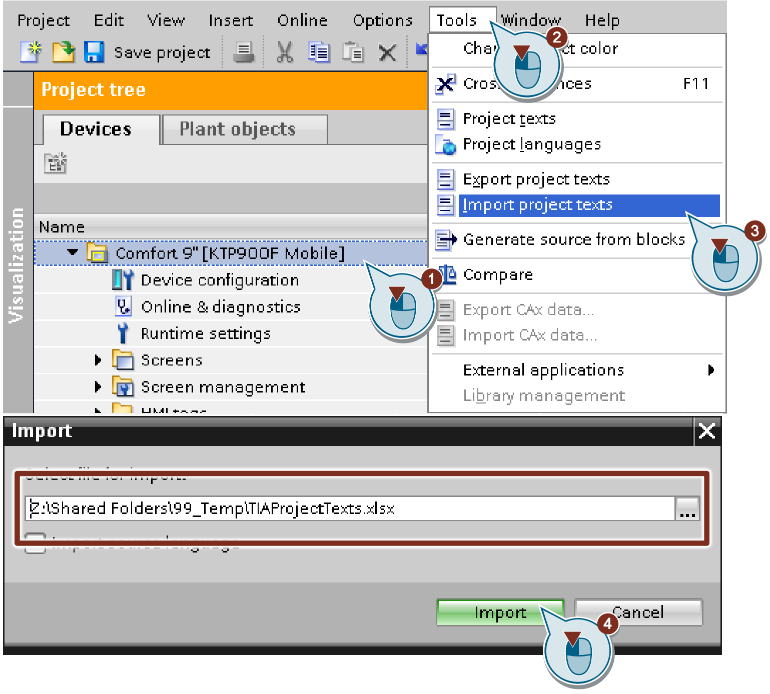
Import

Select file for import:

Z:\Shared Folders\99_Temp\TIAProjectTexts.xlsx

□ Import source language

Figure 2-19: Import general HMI text list



2.11 Operation modes

The operation of a machine can be controlled by several operation modes. The active operation mode of SCP is determined by the value of the 'LSCP_HMIInterface.hmiInterface.operationMode' variable.

NOTE

When operation modes are not required, the default operation mode of the SCP is set to LSCP_OPMODE_FULLOPERATION, which enables the full scope of all SCP functionalities.

The variable is directly written by the built-in support of the operation mode selection provided on the HMI or can be set e.g., by another operation mode manager. Table 2-6 shows the available operation modes.

Table 2-6: Operation modes

Operation mode	PLC constants	Value	Automatic program	Manual control
Full operation	LSCP_OPMODE_ FULLOPERATION	0	✓	✓
Automatic	LSCP_OPMODE_ AUTOMATIC	1	✓	✗
Manual	LSCP_OPMODE_ MANUAL	2	✗	✓
No operation	LSCP_OPMODE_ NOOPERATION	10	✗	✗

'Automatic program' means that it is possible to operate the machine with the default conditions or the home screen (cf. chapter 7). 'Manual control' means that it is possible to jog axes and the kinematics by using the 'Manual Control' screen (cf. chapter 4)

NOTE

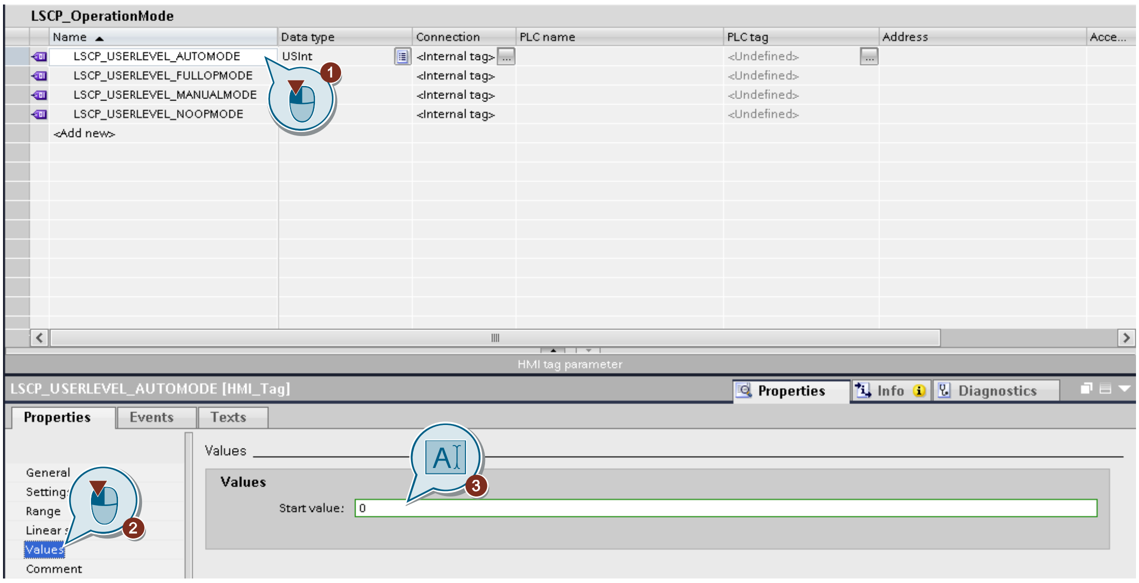
> In the operation mode ‘Manual’ the input ‘motionEnable’ will be evaluated for the motion of axes and kinematics. In the operation mode ‘Full operation’ the ‘motionEnable’ input is only evaluated if the global constant ‘LSCP_MOTION_ENABLE_FULL_OP_MODE’ is set.

User levels for operation mode selection

The selection of operation modes on the HMI, can be configured to require a certain user level. This is done in the HMI tags > ‘LSCP_Tags’ > ‘LSCP_OperationMode’. For each operation mode there is a tag where a required user level for the respective operation mode can be defined. For the values of the user level defined for the SCP refer to chapter 2.6.

Figure 2-20 shows an example for the assignment of the user level for the operation mode ‘Automatic’. The start value of the HMI tag ‘LSCP_USERLEVEL_AUTOMODE’ is set to two. To select the operation mode ‘Automatic’ the operator of the machine needs to be logged in as a user with a user level of two or greater. In the application example these are the user ‘staff’, ‘admin’, and ‘useradmin’.

Figure 2-20: Example of user level assignment for operation modes



LSCP_OperationMode

Name	Data type	Connection	PLC name	PLC tag	Address	Acce...
LSCP_USERLEVEL_AUTOMODE	USInt	<Internal tag>		<Undefined>		
LSCP_USERLEVEL_FULLOPMODE		<Internal tag>		<Undefined>		
LSCP_USERLEVEL_MANUALMODE		<Internal tag>		<Undefined>		
LSCP_USERLEVEL_NOOPMODE		<Internal tag>		<Undefined>		
<Add new>						

LSCP_USERLEVEL_AUTOMODE [HMI_Tag]

Values

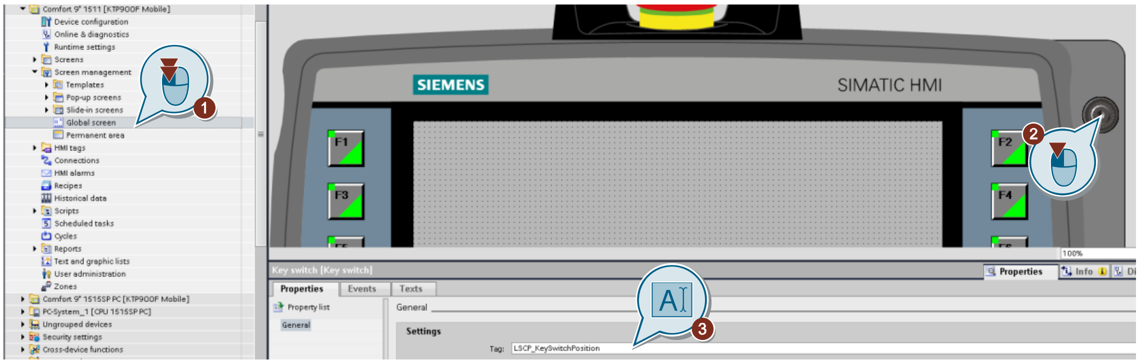
Start value: 0

Key switch integration

When a mobile panel with a key switch is used, it is possible to use the built-in operation mode manager from the SCP. Turning the key switch will open a pop up, where the operator can select a new operation mode.

The assignment of the key switch can be done in the HMI in ‘Screen management’ > ‘Global screen’. The key switch must be connected to the tag ‘LSCP_KeySwitchPosition’ (cf. Figure 2-21).

Figure 2-21: Key switch assignment



Tag: LSCP_KeySwitchPosition

NOTE

The key switch is not working in the HMI runtime simulation. To make use of the provided operation mode manager from the SCP during the simulation, you can assign another variable or a spare function key to the tag ‘LSCP_KeySwitchPosition’. The value of the tag ‘LSCP_KeySwitchPosition’ is Boolean and can be toggled with an invert event in this case.

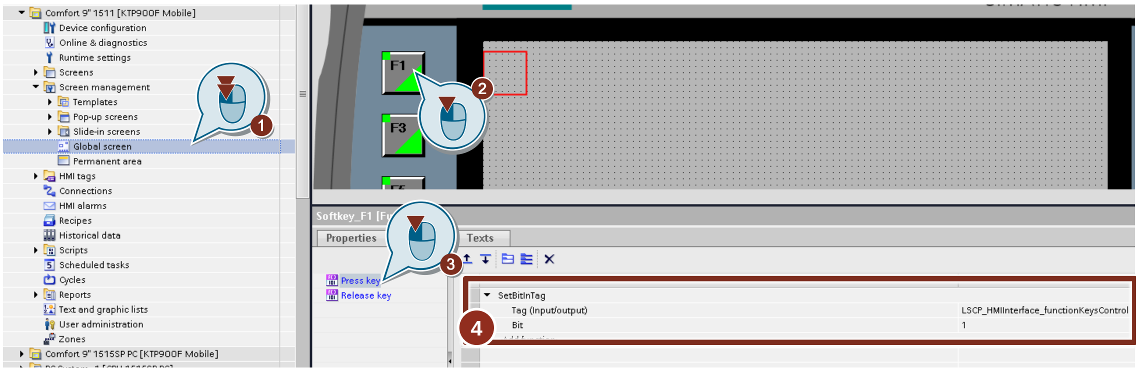
2.12 Function Keys

The function keys of mobile panels can be used to control the program execution of the automatic program (see chapter 7.1). The PLC constant ‘LSCP_USE_HMI_FUNCTION_KEYS’, located in PLC tags > ‘LSCP_Tags’ > ‘LSCP_ProjectConfiguration’, determines the evaluation of function key events and can be activated or deactivated.

Assignment of function keys

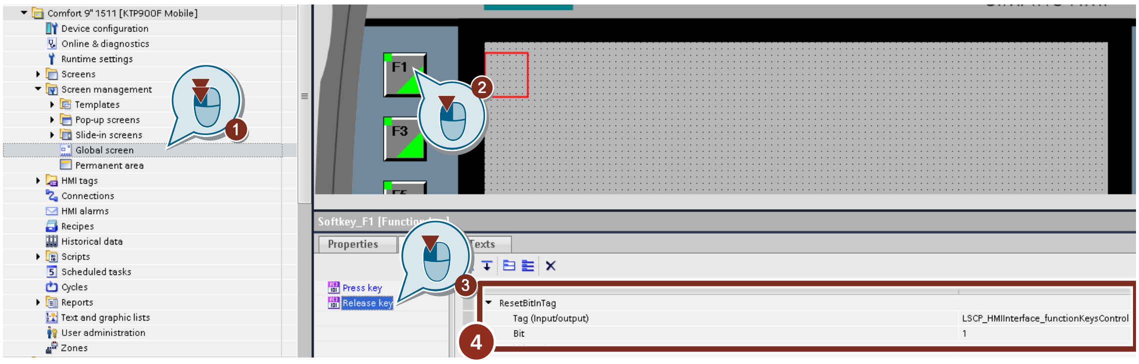
In the application example and the complete panel in the library the function keys are already assigned and activated. The assignment is done in the HMI in ‘Screen management’ > ‘Global screen’. Each button has a press and release event. When using the integration of function keys these events need to be configured as seen in Figure 2-22 and Figure 2-23. The set and reset bits are the same as the number of the function key.

Figure 2-22: Function key press event – F1



SetBitInTag
Tag (Input/output): LSCP_HMIInterface_functionKeysControl
Bit: 1

Figure 2-23: Function key release event – F1



ResetBitInTag
Tag (Input/output) LSCP_HMIInterface_functionKeysControl
Bit 1

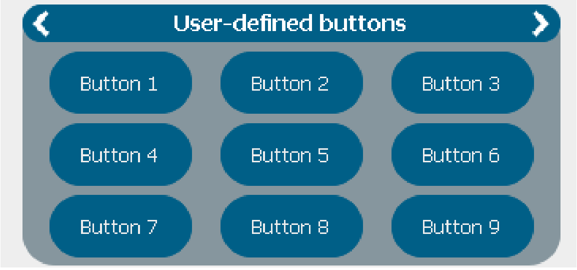
NOTE

In this application example a mobile panel KTP900F is used, that comes with up to 12 physical function keys. It is recommended to use a panel with at least 8 function keys to cover the complete scope of the function key integration.

2.13 User-defined buttons

The Home screen offers up to nine configurable user-defined buttons. The number of visible buttons, the name, and the button appearance (blue or green background color) can be changed for each button individually. The numeration of the buttons starts with one at the first button in the top left corner and ends with nine at the last button in the bottom right corner (cf. Figure 2-24).

Figure 2-24: User-defined buttons in the home screen



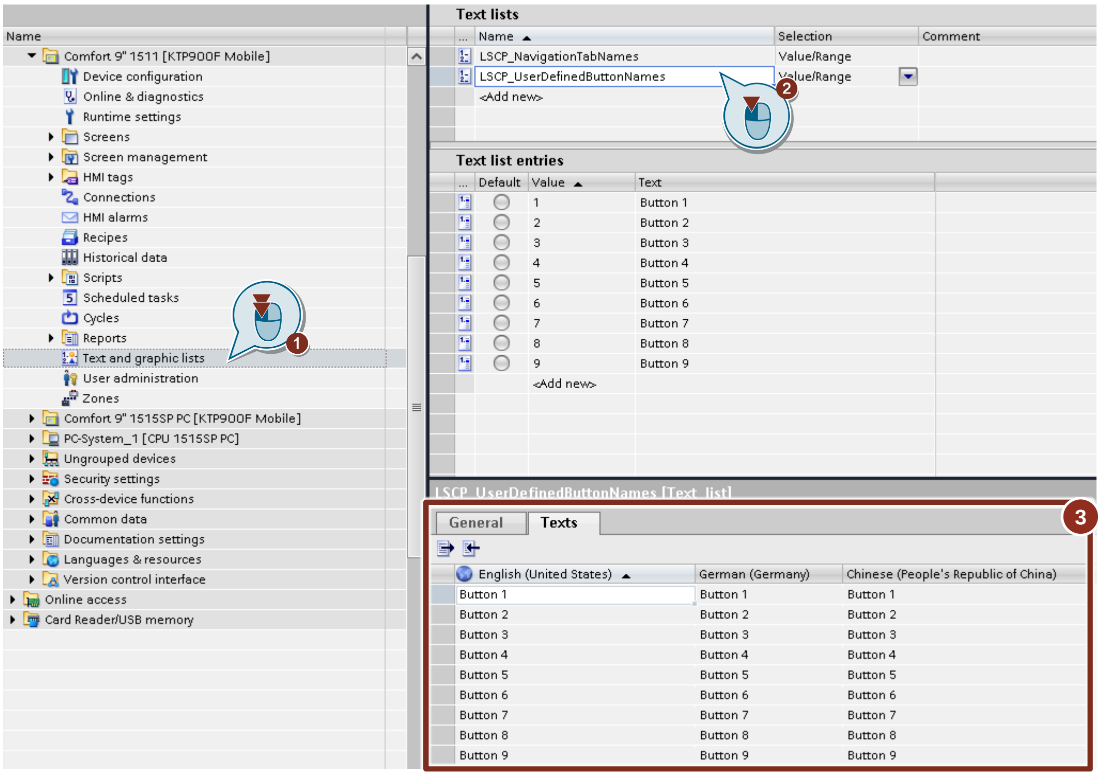
Change number of visible buttons

The number of visible buttons is determined by the PLC constant 'LSCP_NO_OF_HOMESCREEN_BUTTONS', located in PLC tags > 'LSCP_Tags' > 'LSCP_ProjectConfiguration'. A maximum of nine buttons can be used.

Change text of buttons

The button names can be changed in the text list 'LSCP_UserDefinedButtonNames' under 'Text and graphic lists' in the HMI. Each button has its own entry in the text list according to its number. The names can be configured for each language individually (cf. Figure 2-25).

Figure 2-25: Text list entries for user-defined buttons



Text lists

Name	Selection	Comment
LSCP_NavigationTabNames	Value/Range	
LSCP_UserDefinedButtonNames	Value/Range	
<Add new>		

Text list entries

Default	Value	Text
○	1	Button 1
○	2	Button 2
○	3	Button 3
○	4	Button 4
○	5	Button 5
○	6	Button 6
○	7	Button 7
○	8	Button 8
○	9	Button 9
	<Add new>	

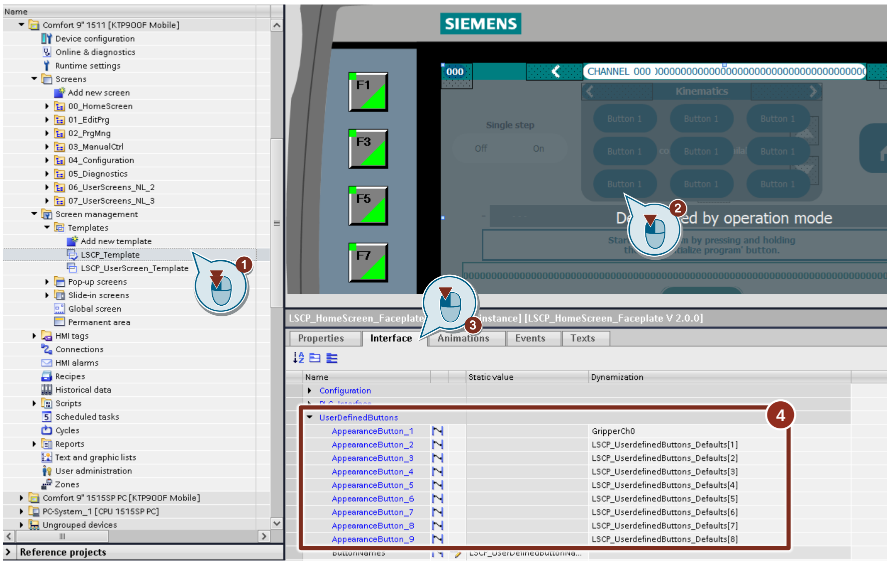
English (United States)	German (Germany)	Chinese (People's Republic of China)
Button 1	Button 1	Button 1
Button 2	Button 2	Button 2
Button 3	Button 3	Button 3
Button 4	Button 4	Button 4
Button 5	Button 5	Button 5
Button 6	Button 6	Button 6
Button 7	Button 7	Button 7
Button 8	Button 8	Button 8
Button 9	Button 9	Button 9

Change appearance of buttons

The button appearance can be changed dynamically between a blue and green background color on the HMI by assigning a Boolean HMI tag to the respective button appearance at the interface of the home screen faceplate. If the value of the connected HMI tag for the appearance of a button is false, the button is colored blue and if the value is true, the button is colored green. The home screen faceplate is in ‘Screen management’ > ‘Templates’ > ‘LSCP_Template’ in the HMI (cf. Figure 2-26).

NOTE The connected variables ‘LSCP_UserdefinedButtons_Default’ to the respective appearance for buttons, are placeholders and can be replaced with own HMI tags.

Figure 2-26: Assignment of appearance for user-defined buttons

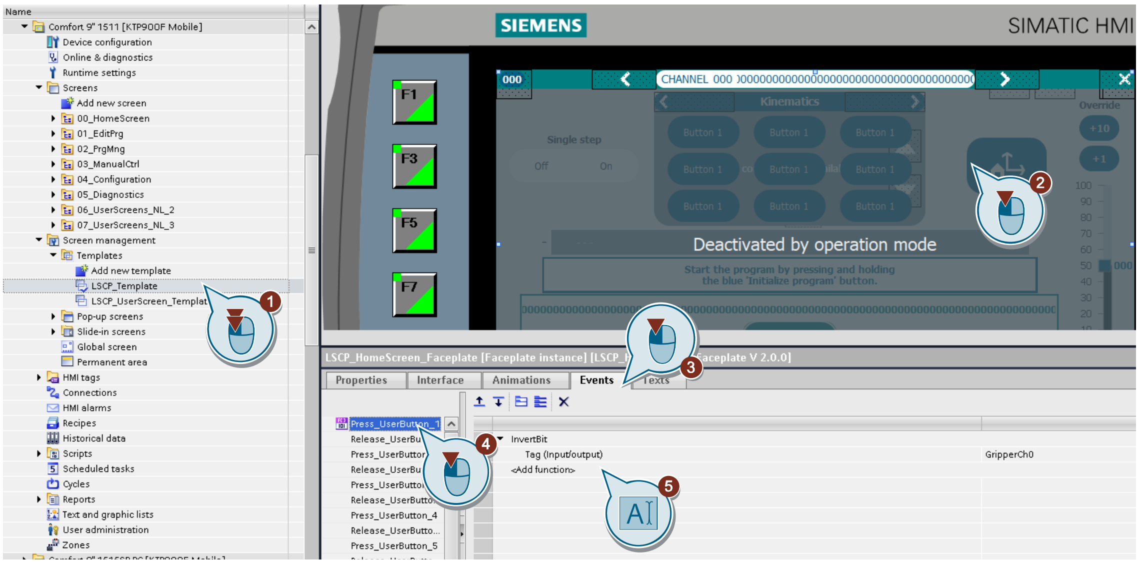


Name	Static value	Dynamization
Configuration		
PLC_Interface		
UserDefinedButtons		
AppearanceButton_1		GripperCh0
AppearanceButton_2		LSCP_UserdefinedButtons_Defaults[1]
AppearanceButton_3		LSCP_UserdefinedButtons_Defaults[2]
AppearanceButton_4		LSCP_UserdefinedButtons_Defaults[3]
AppearanceButton_5		LSCP_UserdefinedButtons_Defaults[4]
AppearanceButton_6		LSCP_UserdefinedButtons_Defaults[5]
AppearanceButton_7		LSCP_UserdefinedButtons_Defaults[6]
AppearanceButton_8		LSCP_UserdefinedButtons_Defaults[7]
AppearanceButton_9		LSCP_UserdefinedButtons_Defaults[8]
ButtonNames	LSCP_UserDefinedButtonNa...	

Assign events to buttons

For each button a press and release event can be configured in the interface of the home screen faceplate (cf. Figure 2-27).

Figure 2-27: Press and release events for user-defined buttons – Button 1



2.14 Keep data consistency between TIA project and PLC retain memory

Programs, points, and configuration data are saved retentively in the DB ‘LSCP_GlobalProjectData’. The start values in the TIA project are not synchronized automatically with the retain memory.

Execute the following steps to synchronize the DB ‘LSCP_GlobalProjectData’ start values (cf. Figure 2-28):

1. Go online.
2. Save a snapshot of the actual values.
3. Copy the snapshot to the start values of the project.

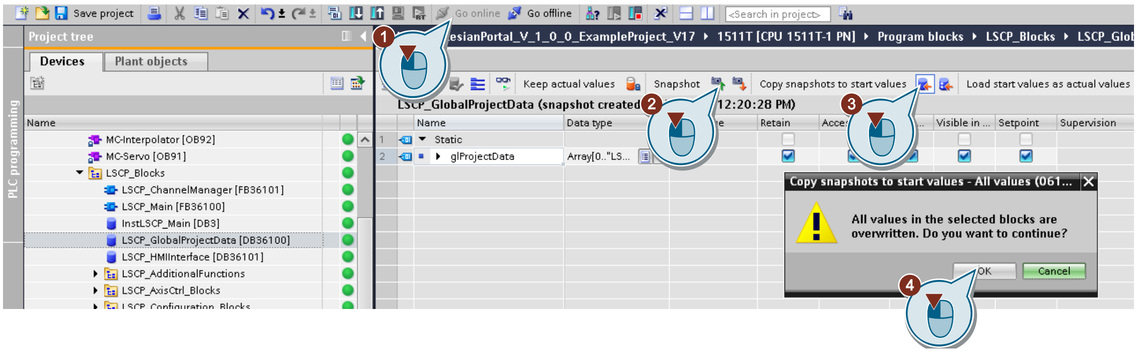
CAUTION

All programs and configurations can be lost after reinitialization at the next download

When you reset the reserved memory, all the variables that were previously in the DB ‘LSCP_GlobalProjectData’ are reinitialized with the start values of the TIA project at the next download.

A compile with the option ‘Reset memory reserve’ should not be executed.
A snapshot of the actual values can be copied to the start values of the TIA project to synchronize the data before the next download.

Figure 2-28: Copy snapshot to start values



Copy snapshots to start values - All values (061...

All values in the selected blocks are overwritten. Do you want to continue?

3 Configuration

3.1 Introduction

This chapter describes the configuration of the different available channels. The following is a list of possible configurations done e.g., by the machine builder:

- In-/Outputs, conditions, and variables
- Message configuration
    - Trigger condition
    - Message texts
- Axes configuration
    - Dynamics
    - Limits, e.g., SW-, HW-, and torque limits
    - General parameter of each axis
- Kinematics configuration
    - Kinematics type
    - Zones
- HMI settings
    - Language setting (English, German, Chinese)
- Stop runtime
- User Login

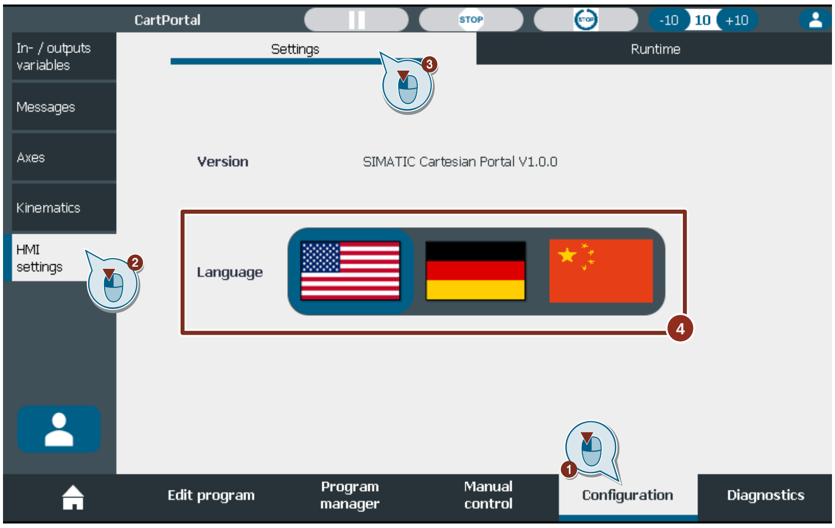
3.2 HMI settings

3.2.1 Select language

Select the desired language in the ‘HMI setting’ > ‘Settings’ section (cf. Figure 3-1).

1. Open the ‘Configuration’ tab in the horizontal bottom navigation.
2. Click on ‘HMI settings’ in the vertical navigation.
3. Click on the ‘Settings’ tab.
4. Select the respective language.

Figure 3-1: Language setting



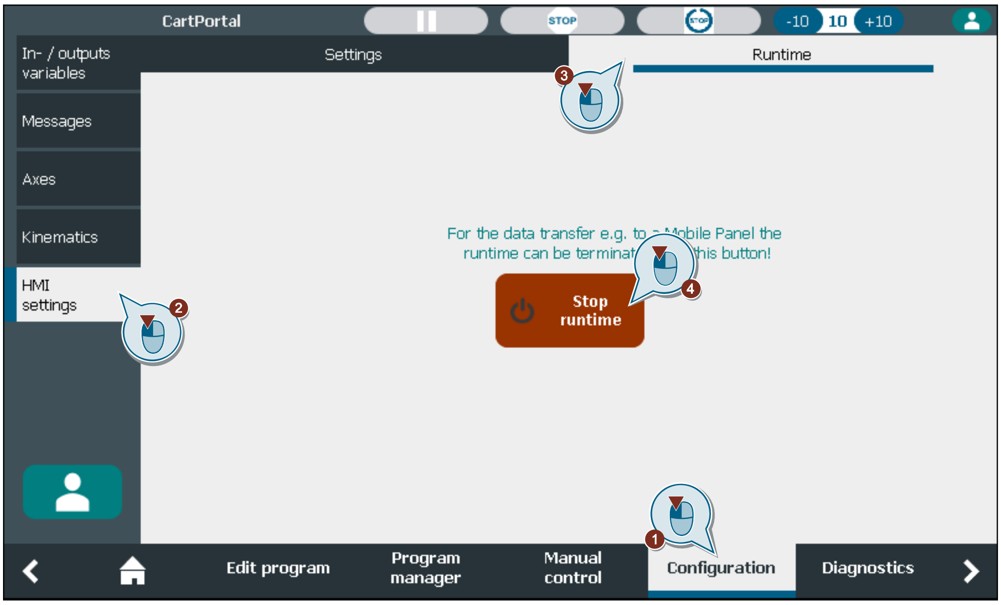
Version SIMATIC Cartesian Portal V1.0.0

3.2.2 Stop the HMI runtime to download a new HMI project

To download a new HMI project the HMI runtime must be stopped, and the transfer mode must be selected. You can stop the HMI runtime in the navigation tab ‘Configuration’ (cf. Figure 3-2).

1. Open the ‘Configuration’ tab in the horizontal bottom navigation.
2. Click on ‘HMI settings’ in the vertical navigation.
3. Click on the ‘Runtime’ tab.
4. Click on the ‘Stop runtime’ button.

Figure 3-2: Stop HMI runtime



For the data transfer e.g. to a Mobile Panel the runtime can be terminated with this button!

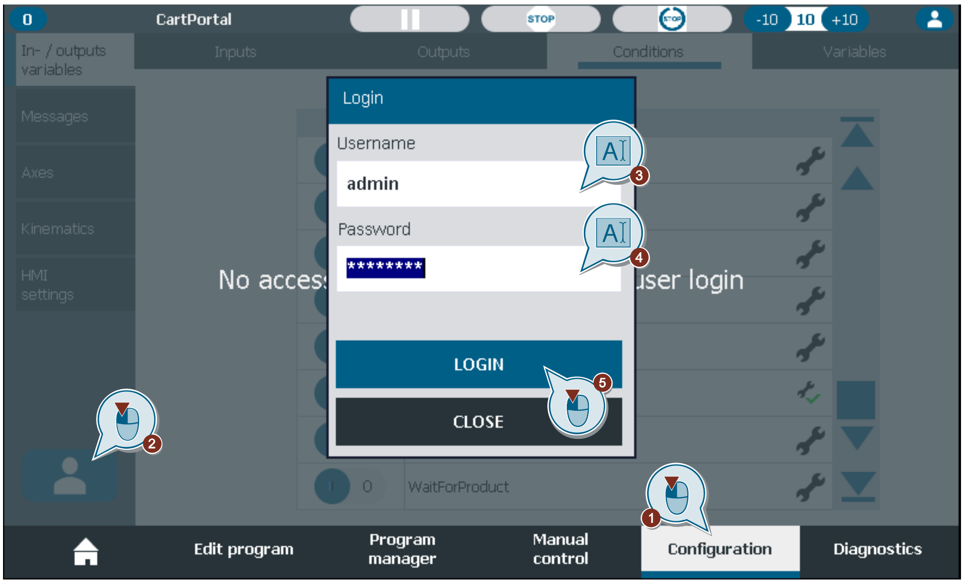
3.3 User login

The user administration is configured in TIA Portal and can only be changed with the user ‘useradmin’. To log in (cf. Figure 3-3):

1. Open the ‘Configuration’ screen in the horizontal bottom navigation.
2. Click on the login button in the vertical navigation or on the cover of the screen.
3. Enter the username.
4. Enter the password.
5. Click on the button ‘Login’.

Afterwards the login symbol will change its color to green to signal a successful login. For more details regarding the user levels and functionalities, see chapter 2.6.

Figure 3-3: User login



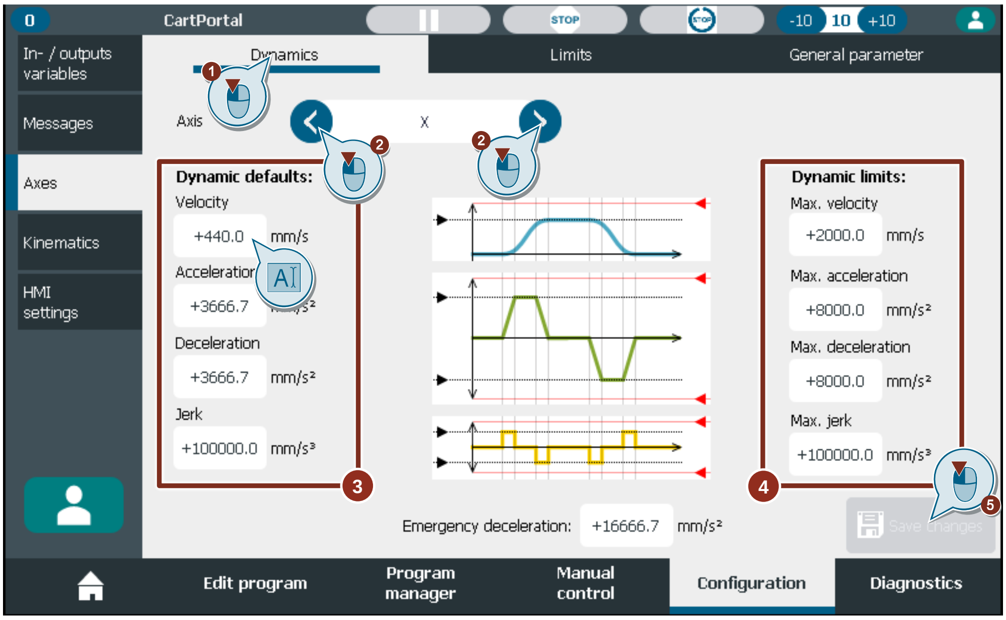
3.4 Axes configuration

3.4.1 Dynamics

Under ‘Dynamics’ the dynamic defaults, dynamic limits and the emergency deceleration can be configured for each axis individually. Changes are stored fail safe in the data block load memory (cf. Figure 3-4).

1. Click on the ‘Dynamics’ tab.
2. Select the desired axis by using the arrow keys.
3. Set the dynamic defaults on the left side.
4. Set the dynamic limits on the right side.
5. Save the changes and accept the warning.

Figure 3-4: Axes dynamics



Dynamic defaults:
Velocity +440.0 mm/s
Acceleration +3666.7 mm/s²
Deceleration +3666.7 mm/s²
Jerk +100000.0 mm/s³

Dynamic limits:
Max. velocity +2000.0 mm/s
Max. acceleration +8000.0 mm/s²
Max. deceleration +8000.0 mm/s²
Max. jerk +100000.0 mm/s³

Emergency deceleration: +16666.7 mm/s²

See chapter 3.6 for an explanation of the warning. Changes are lost when leaving the screen or changing the axis without saving.

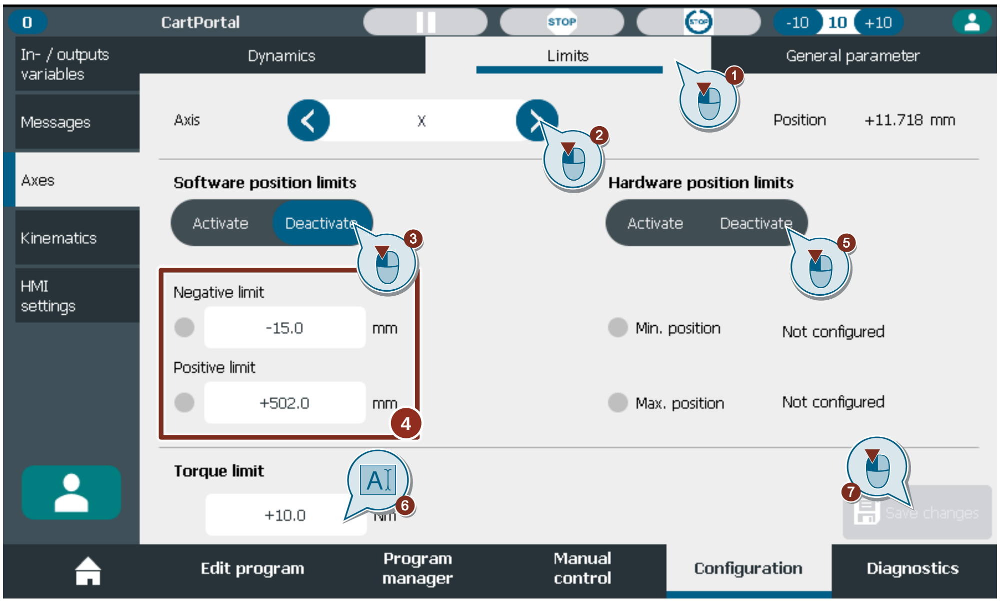
3.4.2 Limits

Under ‘Limits’ the software, hardware, and torque limit can be configured for each axis individually. Changes are stored fail safe in the data block load memory (cf. Figure 3-5).

1. Click on the ‘Limits’ tab
2. Select the desired axis by using the arrow keys.
3. Activate / deactivate the software limits.
4. Set the negative / positive limits.
5. Activate / deactivate the hardware limits. (The tags must be assigned in the TIA-portal).
6. Set the torque limit.
7. Save the changes and accept the warning.

See chapter 3.6 for an explanation of the warning. Changes are lost when leaving the screen or changing the axis without saving.

Figure 3-5: Limit configuration



Axis X
Position +11.718 mm

Software position limits: Activate / Deactivate
Negative limit -15.0 mm
Positive limit +502.0 mm

Hardware position limits: Activate / Deactivate
Min. position Not configured
Max. position Not configured

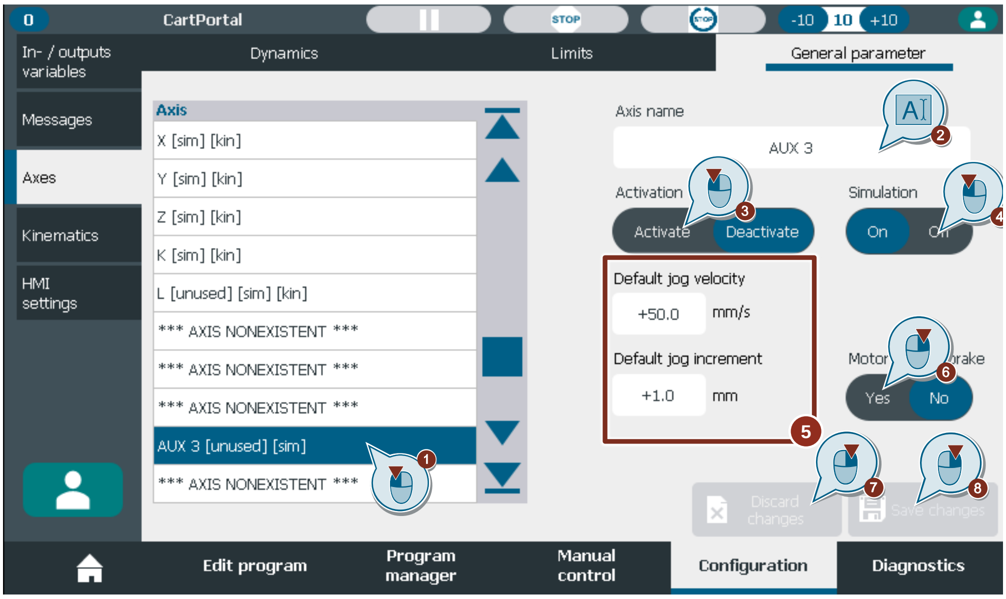
Torque limit +10.0 Nm

3.4.3 General parameter

Under ‘General parameters’ the axis names, the axis simulation mode, the motor holding brake, the jog velocity, the jog increment of each axis individually and the activation of auxiliary axes can be configured (cf. Figure 3-6). Changes are stored fail safe in the data block load memory.

NOTE A change of the axis activation state leads to a reconfiguration of the interpreter module assignment and restart of the interpreter.

Figure 3-6: General parameter



Axis
X [sim] [kin]
Y [sim] [kin]
Z [sim] [kin]
K [sim] [kin]
L [unused] [sim] [kin]
*** AXIS NONEXISTENT ***
*** AXIS NONEXISTENT ***
*** AXIS NONEXISTENT ***
AUX 3 [unused] [sim]
*** AXIS NONEXISTENT ***

Axis name: AUX 3

Default jog velocity: +50.0 mm/s

Default jog increment: +1.0 mm

1. All existing axes are displayed in the table. The square brackets indicate that an axis is not used ‘[unused]’, in simulation ‘[sim]’, a virtual axis ‘[virtual]’ and/or a kinematics axis ‘[kin]’. The text ‘*** AXIS NONEXISTENT ***’ indicates that no axis technology object is connected to this entry. Select the desired axis in the here.
2. Change the axis name.
3. Activate/ deactivate an axis. The kinematic axes cannot be deactivated here. There activation is defined by the kinematics type.
4. Enable/ disable the simulation mode of an axis.
5. Adjust the default jog velocity and default jog increment of an axis.
6. Configure a motor holding brake of an axis.
7. Discard any changes.
8. Save the changes and accept the warning.

See chapter 3.6 for an explanation of the warning. Changes are lost when leaving the screen or changing the axis without saving.

NOTE

> At least one axis must be used per channel.

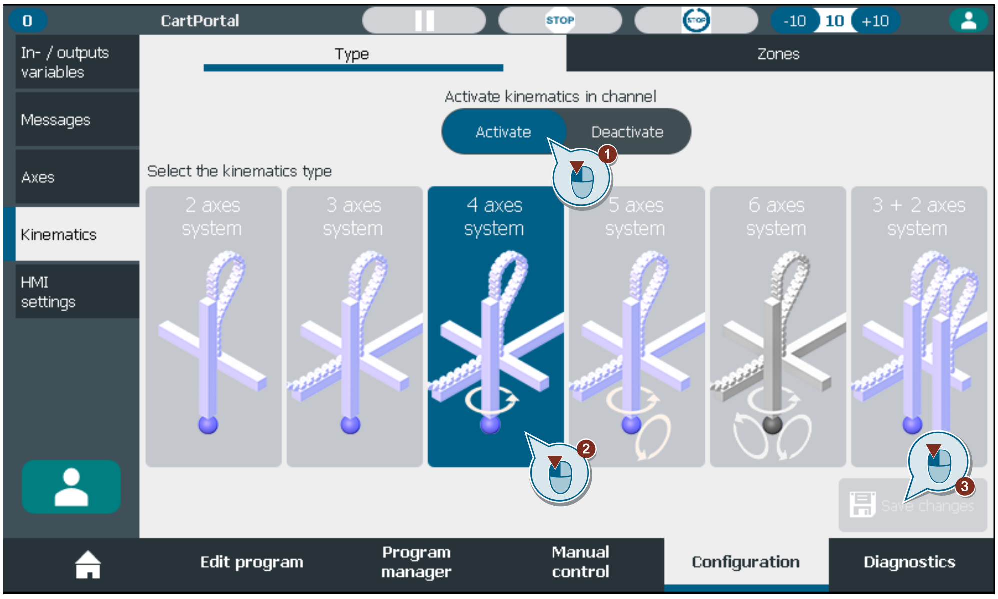
3.5 Kinematics configuration

3.5.1 Kinematics type

Under ‘Type’ the kinematics can be activated or deactivated. If the kinematics is activated the type can be changed (cf. Figure 3-7). Only types with existing axes can be selected and the kinematics cannot be deactivated if no auxiliary axis is activated.

1. Activate or deactivate the kinematics for this channel.
2. Select a kinematics type if the kinematics is activated.
3. Save the changes.

Figure 3-7: Kinematics type



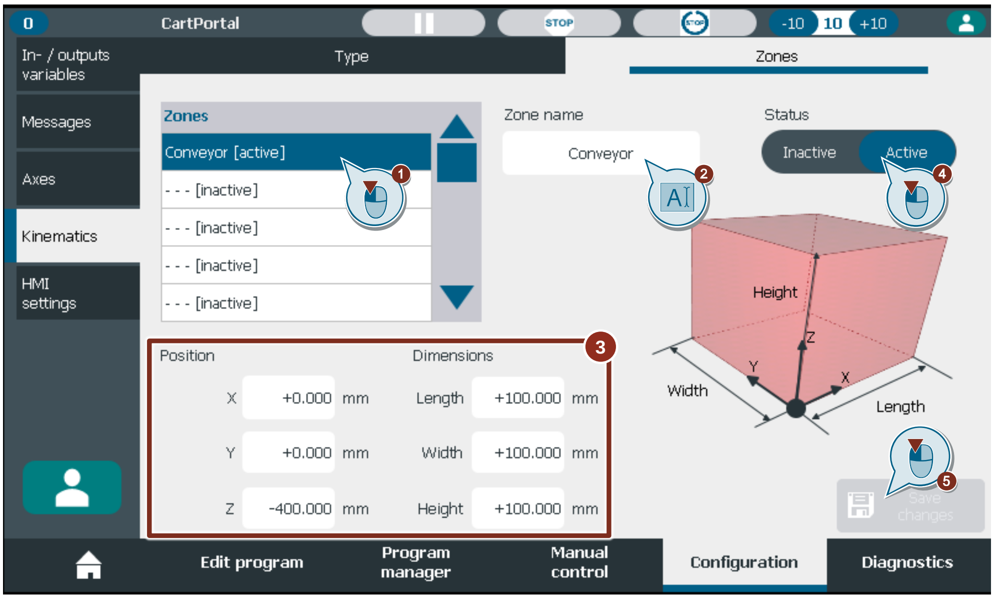
Changes are lost when leaving the screen without saving.

3.5.2 Zones

Under ‘Zones’ blocked cuboid zones can be configured (cf. Figure 3-8).

1. All possible zones are shown in the table. Their state is indicated in the square brackets. Select the zone you want to configure.
2. Enter the name of the zone.
3. Define the position of the zone and the dimensions.
4. Activate/ deactivate the zone.
5. Save the changes.

Figure 3-8: Zone configuration



Position			Dimensions		
X	+0.000	mm	Length	+100.000	mm
Y	+0.000	mm	Width	+100.000	mm
Z	-400.000	mm	Height	+100.000	mm

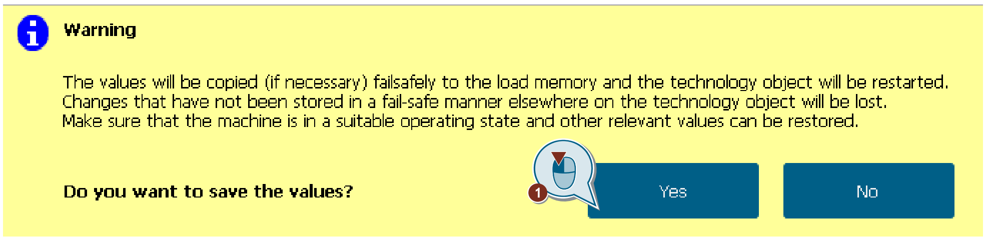
CAUTION	**Zone collision evaluation** A zone collision triggers a fast stop when the zone collision happens. Consider the stopping distance of the kinematics when configuring a blocked zone.

3.6 Writing to the online load memory of the technology object data block

The following warning message indicates that the changes of the technology object parameters are written to the load memory of the technology object data block. Subsequently, a restart of the technology object is performed, if necessary (cf. Figure 3-9).

NOTICE	**Possible data loss** Changes that have not been stored in a fail-safe-manner elsewhere on the technology object will be lost. This is not relevant if the application is used exclusively.

Figure 3-9: Alert message -writing to load memory



Warning

The values will be copied (if necessary) failsafely to the load memory and the technology object will be restarted.
Changes that have not been stored in a fail-safe manner elsewhere on the technology object will be lost.
Make sure that the machine is in a suitable operating state and other relevant values can be restored.

Do you want to save the values?

Make sure, that these online program changes are synchronized to the offline program before downloading other changes to the device e.g., by synchronizing the changes before downloading or manual synchronization (compare offline / online). For further information see chapter 2.3.2.

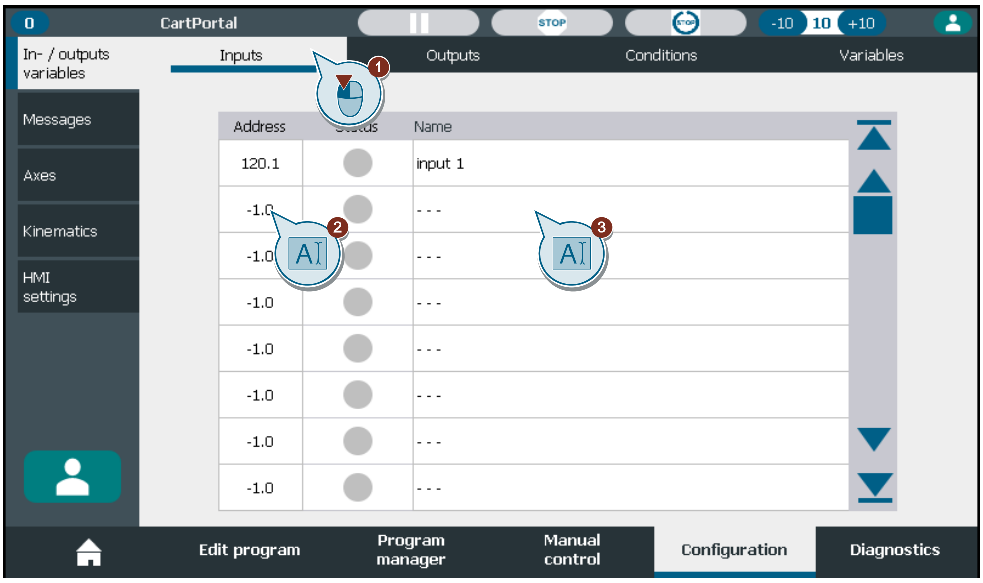
3.7 Inputs, Outputs, conditions, and variables

3.7.1 Input configuration

Digital inputs can be configured directly on the HMI. The I/O addresses from the DI module and the wiring diagram must be known. The actual status is indicated by the circle with grey meaning ‘FALSE’ and blue meaning ‘TRUE’ (cf. Figure 3-10).

1. Click on the ‘Inputs’ tab.
2. Enter the address of the digital input.
3. Enter the name of the digital input.
4. Delete an input by deleting its name.

Figure 3-10: Input configuration



Address	Status	Name
120.1	●	input 1
-1.0	●	- - -
-1.0	●	- - -
-1.0	●	- - -
-1.0	●	- - -
-1.0	●	- - -
-1.0	●	- - -
-1.0	●	- - -

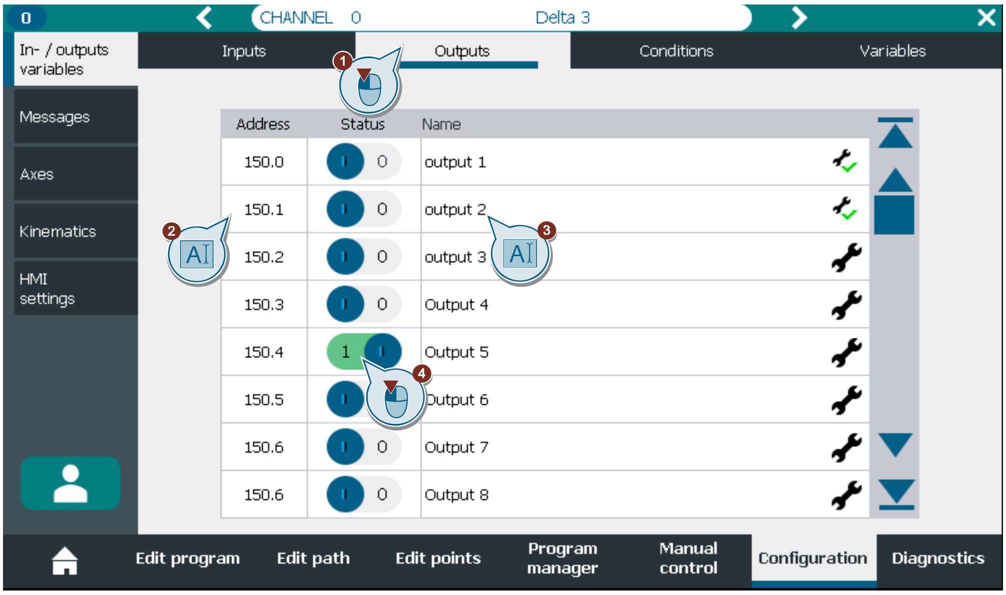
NOTE To simulate an input via a watch table in the TIA Portal a free address which is not configured in any I/O module can be configured or the force table can be used.

3.7.2 Output configuration

Digital outputs can be configured directly on the HMI. The I/O addresses from the DQ module and the wiring diagram must be known (cf. Figure 3-11).

1. Click on the ‘Outputs’ tab.
2. Enter the address of the digital output.
3. Enter the name of the digital output.
4. Set the state of the output manually.
5. Delete an output by deleting its name.

Figure 3-11: Output configuration



Address	Status	Name
150.0	I 0	output 1
150.1	I 0	output 2
150.2	I 0	output 3
150.3	I 0	Output 4
150.4	1 I	Output 5
150.5	I	Output 6
150.6	I 0	Output 7
150.6	I 0	Output 8

3.7.3 Condition configuration

Predefined conditions

The following variables are predefined. You must not change them. You can use them for programming (cf. chapter 6).

Table 3-1: Predefined Conditions

Conditions	Description
INITIALIZE program	As soon as the condition 'INITIALIZE Program' is set, the commands following the jump label ***INITIALZE POGRAM*** are processed. The end of the initialization program is marked by the command 'Wait for START program'.
START program	If the initialize program commands are finished and the command 'Wait for START program' is waiting, the condition 'START program' will continue the automatic program.
PAUSE program	Setting this condition will pause the program.
CONTINUE program	A paused program can be continued with the condition 'CONTINUE program'.
STOP program	You can stop the program if you set this variable to TRUE. The kinematics and the axes will stop with the set dynamics.
FAST STOP program	The condition 'FAST STOP program' can be used to stop the automatic program with maximum dynamics in an emergency, such as triggering an emergency stop button or a light curtain.
CYCLE STOP program	To stop the program execution at a predefined command the 'CYCLE STOP' can be used. The condition is set when the 'CYCLE STOP program' button on the homescreen or in the headbar is pressed.

NOTE

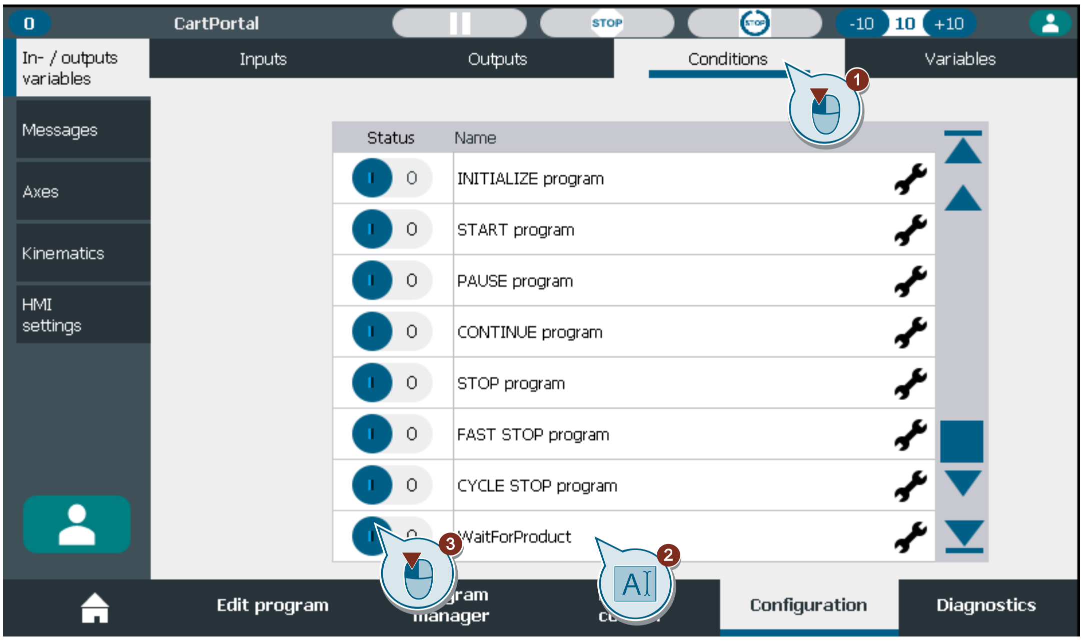
The program execution can be controlled by interacting with these conditions (cf. chapter 7.1.4) or with the buttons on the home screen or the head bar.

User defined conditions

To create or adapt a condition (cf. Figure 3-12):

1. Click on the 'Conditions' tab.
2. Enter a name for the condition.
3. Set the state of the output manually.
4. Delete an output by deleting its name.

Figure 3-12: User defined condition



Status	Name
I 0	INITIALIZE program
I 0	START program
I 0	PAUSE program
I 0	CONTINUE program
I 0	STOP program
I 0	FAST STOP program
I 0	CYCLE STOP program
I 0	WaitForProduct

3.7.4 Link inputs and conditions to outputs or conditions with AND/OR logic

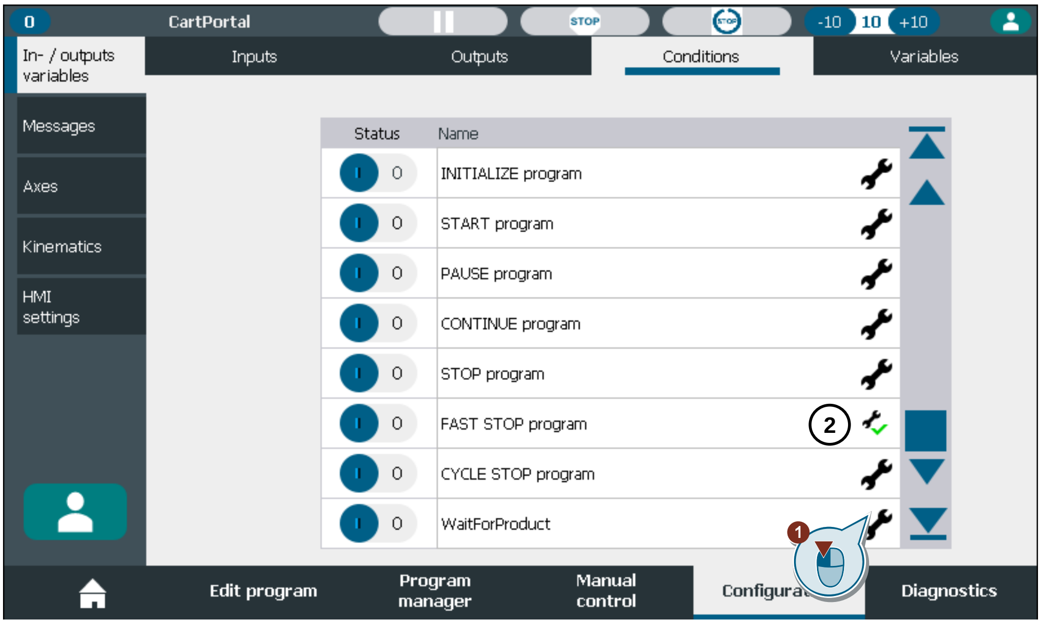
A wait or jump condition often depends on a combination of several inputs or conditions. To avoid programming of multiple wait commands in a row a condition or output can be defined as the result of AND/OR-logic depending on inputs or other conditions (cf. Figure 3-13). The result state of the condition will be updated cyclically in the background.

NOTE

> You can link configured conditions which are the result of linked signals. So, you can split and link different operations instead of linking a lot of single signals to one condition.

1. Click on the wrench icon to open the logic configuration of a condition or output
2. A configured condition is indicated by a green checkmark next to the wrench.

Figure 3-13: User defined condition – logic configuration overview



Status	Name
I 0	INITIALIZE program
I 0	START program
I 0	PAUSE program
I 0	CONTINUE program
I 0	STOP program
I 0	FAST STOP program ✓
I 0	CYCLE STOP program
I 0	WaitForProduct

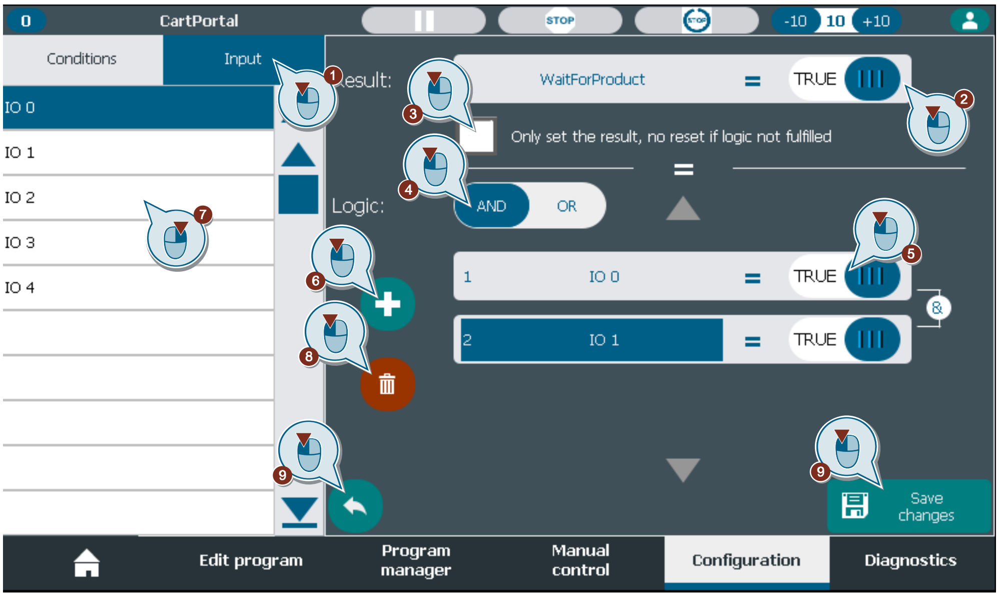
In the logic configuration you can do the following (see Figure 3-14):

1. Select inputs or conditions in the selection table on the left side
2. Specify what the result of the condition should be if the connected logic is fulfilled.
3. If the checkbox is set:
    a. Condition = TRUE: condition is set if the logic is fulfilled, but not reset if the logic is not fulfilled.
    b. Condition = FALSE: condition is reset if the logic is fulfilled, but not set if the logic is not fulfilled.

    If the checkbox is not set:

    c. Condition = TRUE: condition is set if the logic is fulfilled and reset if the logic is not fulfilled.
    d. Condition = FALSE: condition is reset if the logic is fulfilled and set if the logic is not fulfilled.
4. Define how the signals are logically connected - AND or OR.
5. Signal is linked with level TRUE or FALSE.
6. Add another signal.
7. After adding another signal, click on a signal in the table to link it to the logic.
8. Press the delete button to delete the signal which is currently highlighted.
9. Save the configuration of the logic (right) or discard the changes and leave the screen (left).

Figure 3-14: User defined condition – logic configuration – example 1



Examples

Figure 3-14 shows the operation

```
WaitForProduct = (IO 0 = TRUE) AND (IO 1 = TRUE)
```

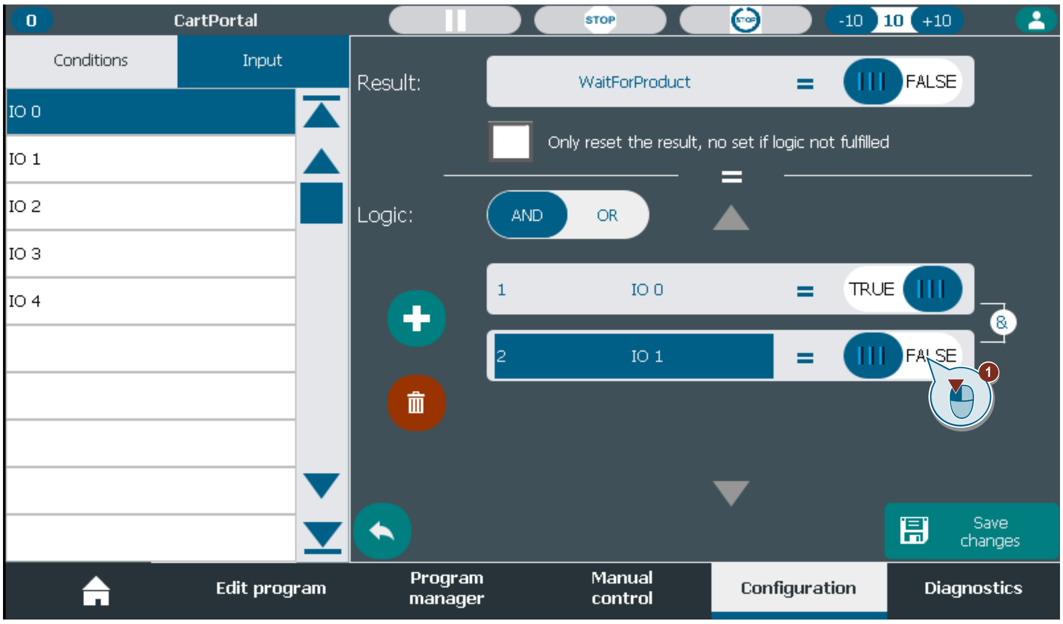
The result of the AND-operation will be inverted if the result state is set to FALSE (cf. Figure 3-14 – (2)).

```
WaitForProduct = NOT ((IO 0 = TRUE) AND (IO 1 = TRUE))
```

If you set a logic condition to FALSE, the logic state will be inverted before the operation (cf. Figure 3-15).

```
WaitForProduct = NOT ((IO 0 = TRUE) AND (IO 1 = FALSE))
```

Figure 3-15: User defined condition – logic configuration – example 2



Result: WaitForProduct = FALSE

□ Only reset the result, no set if logic not fulfilled

Logic: AND OR

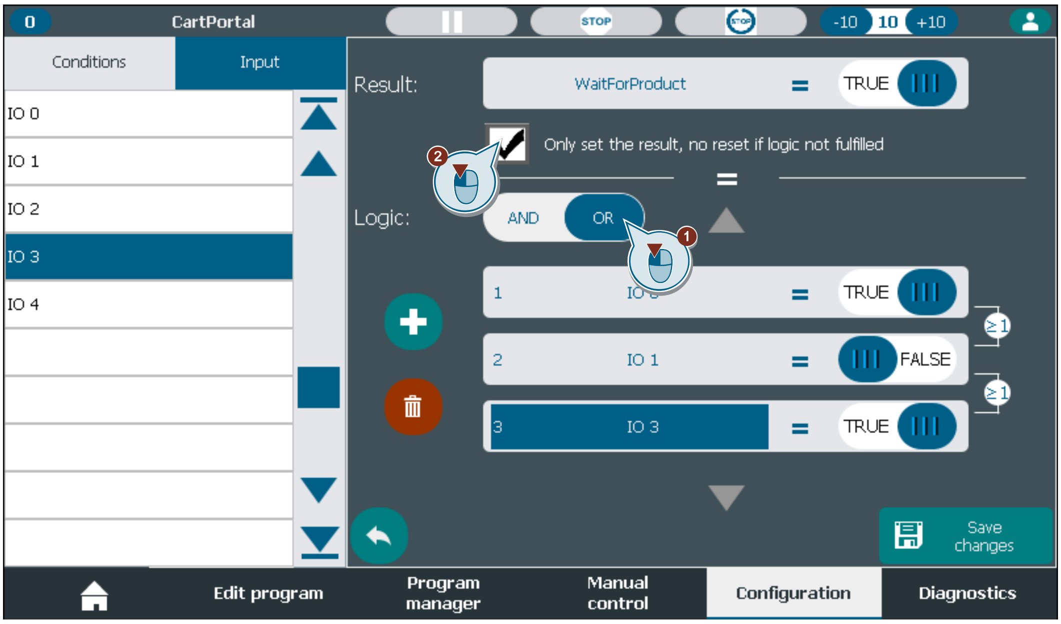
1 IO 0 = TRUE

2 IO 1 = FALSE

The logic can also be evaluated with OR logic and with the addition of ‘Only set the result, no reset if logic is fulfilled’. The logic in Figure 3-16 evaluates to

```
IF ((IO 0 = TRUE) OR (IO 1 = FALSE) OR (IO 3 = TRUE)) THEN
   WaitForProduct = TRUE
END_IF
```

Figure 3-16: User defined condition – logic configuration – example 3



Result: WaitForProduct = TRUE

☑ Only set the result, no reset if logic not fulfilled

Logic: AND OR

1 IO 0 = TRUE

2 IO 1 = FALSE

3 IO 3 = TRUE

3.7.5 Variable configuration

Predefined variables

The axes positions (kinematics axes / auxiliary axes) are already predefined as floating-point variables on the HMI and must not be changed. The values will be updated cyclically in the background. The axes positions can be used to set a command at a specific position of an axis, e.g., set an output. Every connected valid axis position (cf. chapter 2.8) is connected to one variable.

⚠ CAUTION

Variables will be overwritten or there will be unused variables when changing the number of valid connected axes.

The predefined variables and the user defined variables are saved in the same array.

When the number of connected axes is changed the number of predefined variables is also changed. When more axes are connected user defined variables can be overwritten. In general commands that refer to these predefined variables can be connected to the wrong variable after the change since the command only refers to the index in the variable array This could lead to unexpected behavior during the program execution.

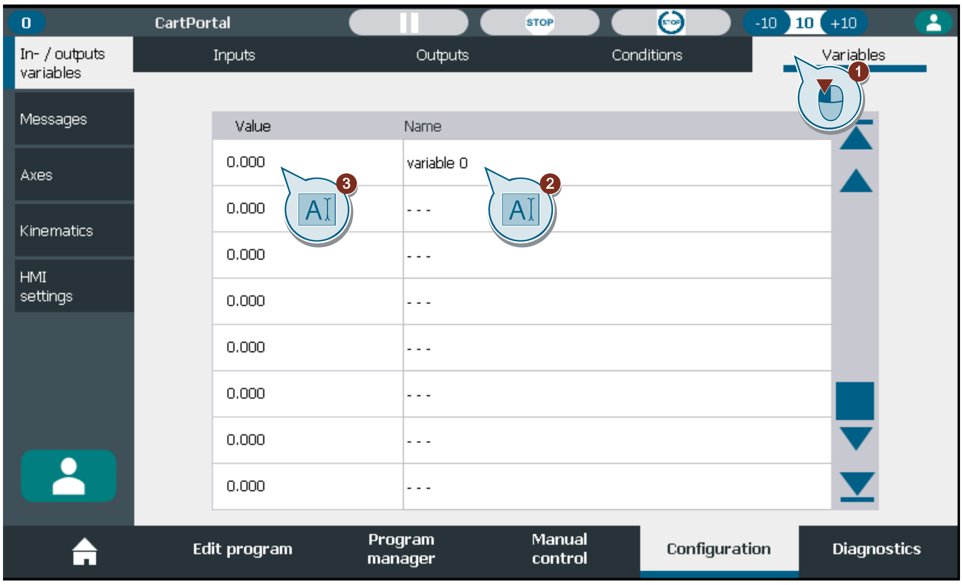
Check the variables and their use in the program after changing the interconnected axes.

User defined floating-point variables

You can configure floating-point variables e.g., as counter variables (cf. Figure 3-17).

1. Click on the ‘Variables’ tab.
2. Enter a name for the variable.
3. Enter a floating-point value for the variable.
4. Delete a variable by deleting its name.

Figure 3-17: Variables configuration



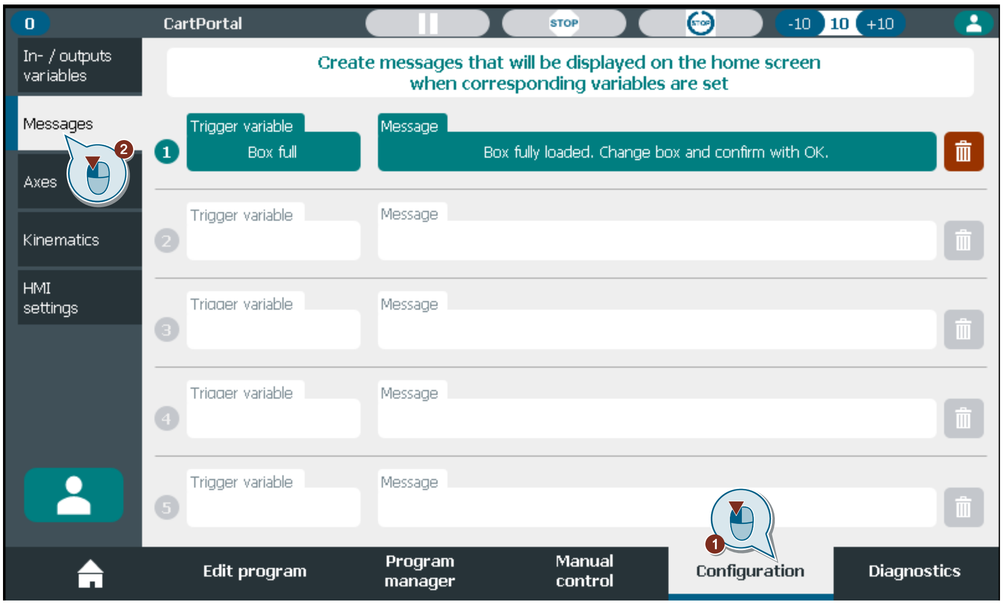
Value	Name
0.000	variable 0
0.000	- - -
0.000	- - -
0.000	- - -
0.000	- - -
0.000	- - -
0.000	- - -
0.000	- - -

3.8 Messages

Up to five messages can be created and linked to a condition. As soon as the condition is 'TRUE', the associated message is displayed on the home screen. If the condition is set to ‘FALSE’, the message disappears. To enter the messages screen (cf. Figure 3-18):

1. Open the ‘Configuration’ tab in the horizontal bottom navigation.
2. Click on ‘Messages’ in the vertical navigation.

Figure 3-18: Message configuration



Create messages that will be displayed on the home screen when corresponding variables are set

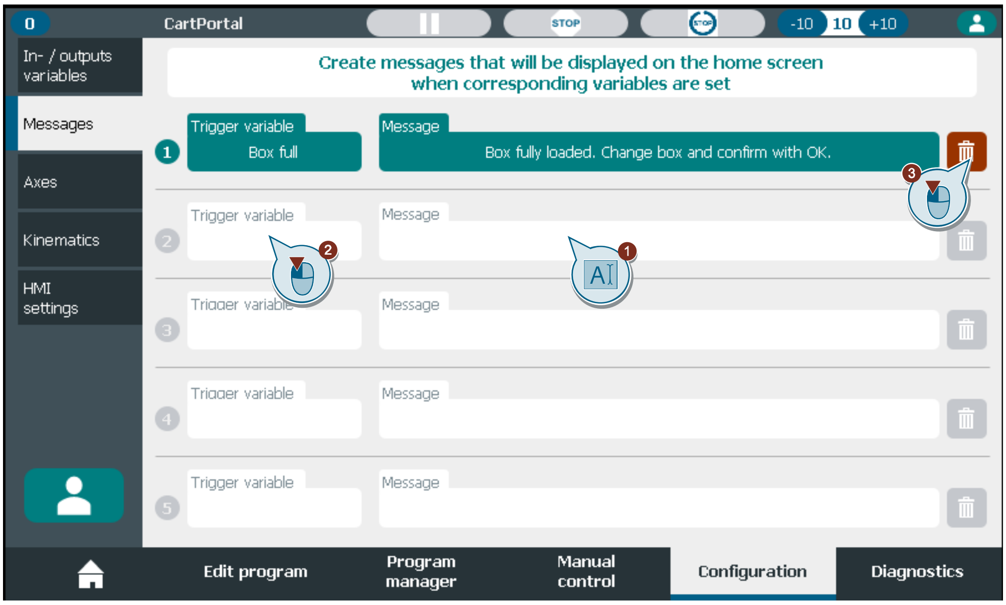
1 Trigger variable: Box full — Message: Box fully loaded. Change box and confirm with OK.

3.8.1 Message configuration

Under ‘Messages‘ up to five messages can be configured (cf. Figure 3-19).

1. Enter a message text.
2. Connect the trigger of the message to a configured condition.
3. Delete the trigger condition.

Figure 3-19: Message configuration



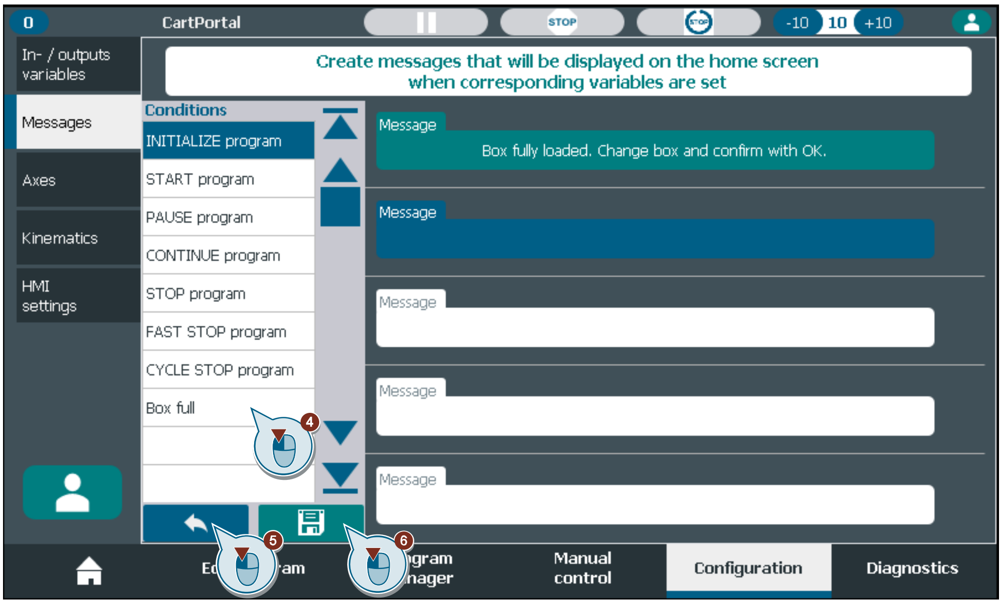
Create messages that will be displayed on the home screen when corresponding variables are set

1 Trigger variable: Box full — Message: Box fully loaded. Change box and confirm with OK.

4. Chose the trigger condition.

5. Discard the selection.
6. Save the selection.
7.

Figure 3-20: Message on home screen



Create messages that will be displayed on the home screen when corresponding variables are set

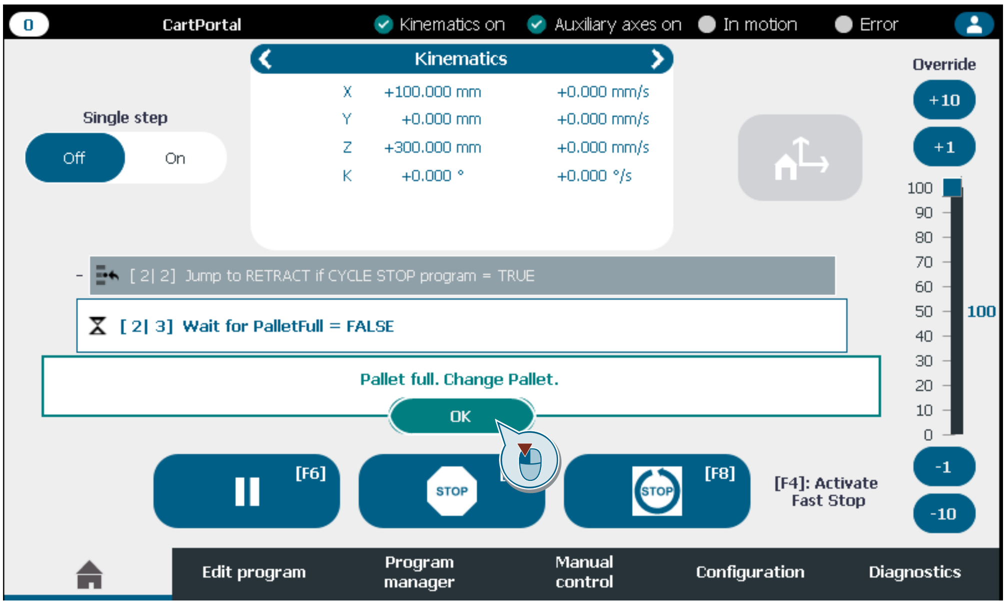
- INITIALIZE program
- START program
- PAUSE program
- CONTINUE program
- STOP program
- FAST STOP program
- CYCLE STOP program
- Box full

Box fully loaded. Change box and confirm with OK.

3.8.2 Message display

If the trigger condition of a message is ‘TRUE’ the message is displayed on the home screen (cf. Figure 3-21).

Figure 3-21: Message display on home screen



X	+100.000 mm	+0.000 mm/s
Y	+0.000 mm	+0.000 mm/s
Z	+300.000 mm	+0.000 mm/s
K	+0.000 °	+0.000 °/s

- [2| 2] Jump to RETRACT if CYCLE STOP program = TRUE

[2| 3] Wait for PalletFull = FALSE

Pallet full. Change Pallet.

The message disappears if the trigger condition is ‘FALSE’. Acknowledging the message on the home screen by clicking ‘OK’ also resets the trigger condition of that message.

4 Manual control

The screen ‘Manual control’ offers the following possibilities:

Kinematics and auxiliary axes control:

- Enable / disable single axis or all axes at once.
- Open / close the motor holding brake if existent.
- Jog axis (incremental / continuous) or jog to a target position.
- Home axis (active homing / absolute encoder adjustment / direct homing).
- Specification of jog velocity and increment.
- Change axis override.

Point table:

- Manually configure a point.
- Teach a point by moving the kinematics axis by axis.

4.1 Enable/ Disable axes, open motor holding brake and acknowledge errors

Enable axes

You must enable the axes before you can jog or execute active homing. You must enable all kinematics axes to teach or approach a point (cf. Figure 4-1).

1. Select an axis with the arrow buttons.
2. Click ‘Enable’ (single axis) to enable the currently selected axis.
3. Click ‘Enable’ (all axes) to enable all used axes.

Disabling the axes

4. Click ‘Disable’ (single axis) to disable the currently selected axis.
5. Click ‘Disable’ (all axes) to disable all enabled axes.

Open motor holding brake

6. Click ‘Open’ to open the motor holding brake of the selected axis. You can open the motor holding brake when the axis is disabled. You can open the motor holding brakes of multiple axes at the same time. A motor holding brake must be configured (cf. chapter 3.4.3).
7. Click ‘Close’ to close the motor holding brake of the selected axis.

If you enable an axis the motor holding brake will be opened as well, but the control is active.

NOTICE

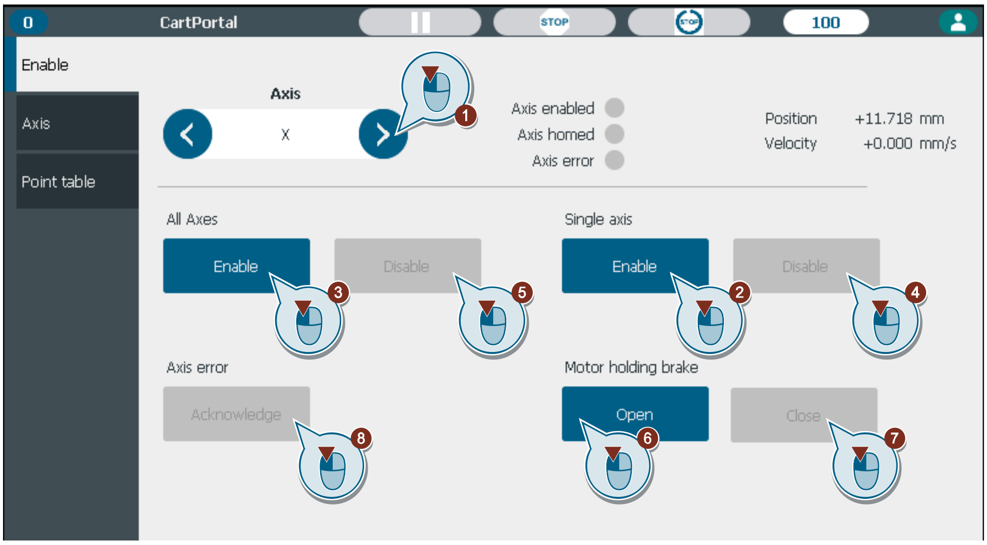
Vertical axis can sack if you open the brake

Fasten the axis before you open the motor holding brake if a danger of sacking or self-powered movement is given.

Acknowledge error

8. Press the ‘Acknowledge’ button to acknowledge all axis errors. Details on the axis error can be seen in the Diagnostics tab (cf. chapter 8). The user login must be sufficient to acknowledge errors (cf. chapter 2.6).

Figure 4-1: Enabling the axes



If an automatic program is paused axes that are not enabled outside of the ‘Manual control’ tab can be enabled. The ‘Enable’ (all axes) button enables only axes which are not enabled already. The ‘Disable’ (all axes) only disables axes that were enabled by the ‘Manual control’ tab.

4.2 Home axis

To navigate to the ‘Home’ tab:

1. Click on the ‘Axis’ tab.
2. Click on the ‘Home’ tab.
3. Select the axis for which the homing should be executed.

The ‘Home’ tab offers three homing modes depending on the currently active encoder of the selected axis. Table 4-1 shows an overview of the available homing modes with regards to the active encoder.

Table 4-1: Homing modes overview

Homing mode	Incremental encoder	Absolute encoder
Absolute encoder adjustment (mode 7)		x
Active homing (mode 3)	x	
Direct homing (mode 0)	x	x

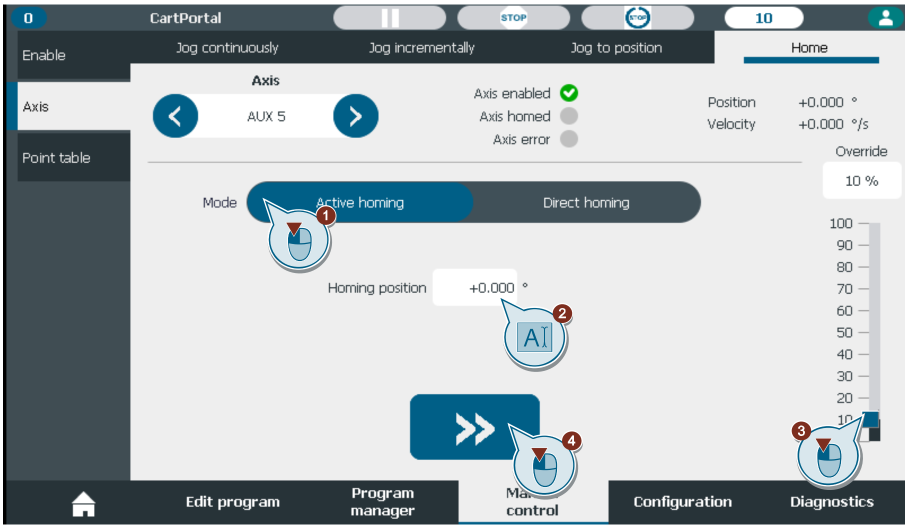
Active homing

A positioning axis performs a homing movement according to the technology object configuration. After the completion of the motion, the axis is homed to the homing position. The active homing mode is only available with incremental encoders. The axis must be enabled for this homing mode (cf. Figure 4-2).

1. Select ‘Active homing’.
2. Enter the home position for the axis.
3. Select the override. The override affects the homing and approach velocity configured in the TO configuration in the TIA Portal.

4. Press and hold the homing button until the active homing sequence is finished.

Figure 4-2: Active homing



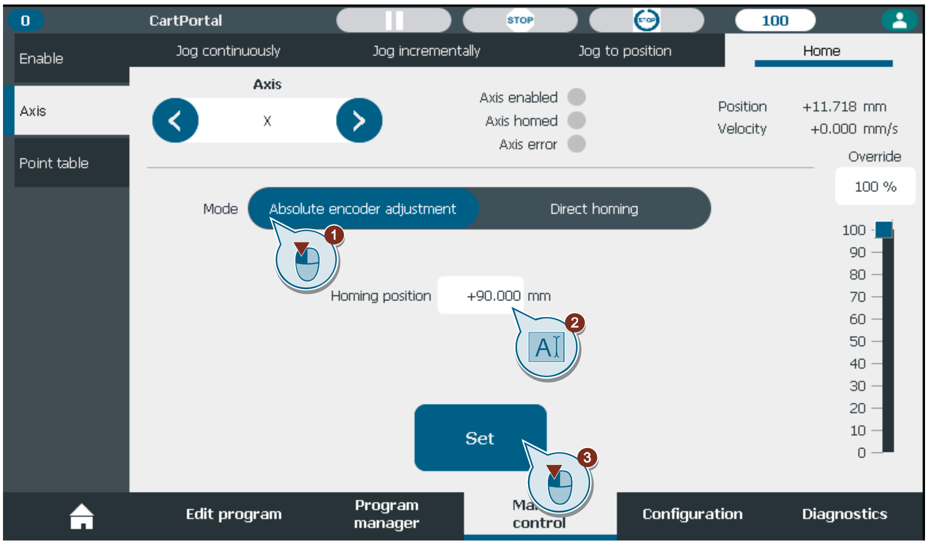
Absolute encoder adjustment

The actual position of the selected positioning axis is set to the homing position. The calculated absolute encoder offset is stored retentively in the CPU. You do not have to home the axes again after power on of the PLC. This mode is only available with absolute encoders (cf. Figure 4-3).

1. Select ‘Absolute encoder adjustment’.
2. Enter the homing position.
3. Click the ‘Set’ button.

NOTICE	**Absolute encoder adjustment of axis in simulation** If an axis, which is already homed via absolute encoder adjustment, is later set to simulation mode, the reference position will normally not match. Executing another absoluter encoder adjustment will result in overwriting of the absolute encoder offset. Disabling the simulation mode will then require another absolute encoder adjustment since the absolute encoder offset was overwritten. It is recommended to use direct homing with simulated axes.

Figure 4-3: Absolute encoder adjustment



Position +11.718 mm
Velocity +0.000 mm/s
Override 100 %
Mode Absolute encoder adjustment
Homing position +90.000 mm

Direct homing

The actual position of the selected positioning axis is set to the homing position. The homing position will be lost after a power off/power on of the PLC and a direct homing must be executed again. Therefore, this mode is only recommended for testing or for axes with incremental encoders with no homing mark or reference cam. Direct homing is available for axes with incremental and absolute encoders. The operation on the HMI is like the absolute encoder adjustment.

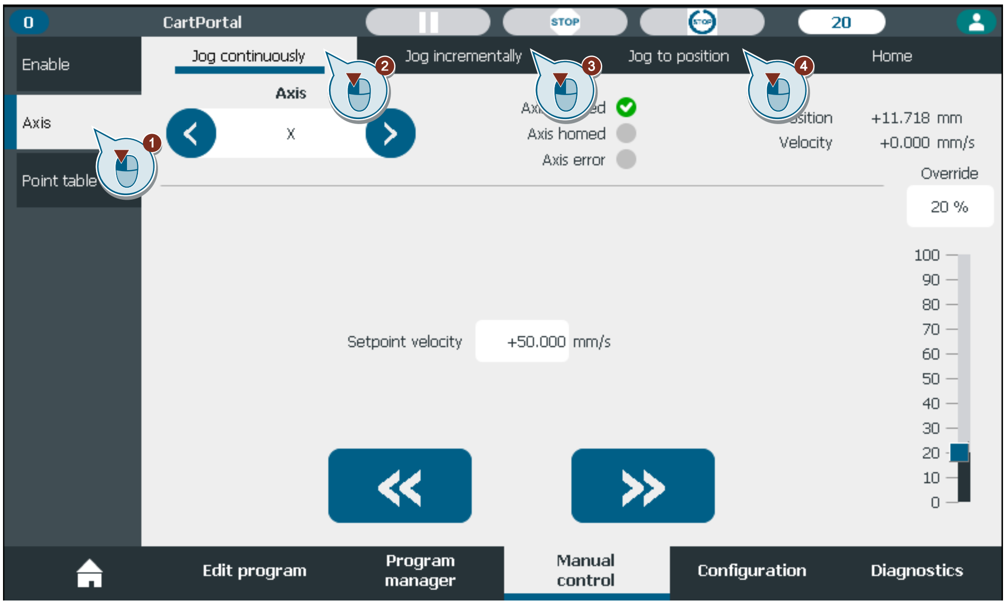
4.3 Jog axes

SIMATIC Cartesian Portal offers three modes to jog an individual axis. To navigation to the jog tabs (cf. Figure 4-4):

1. Select the ‘Axis’ tab.
2. Select the ‘Jog continuously’ tab or
3. Select the ‘Jog incrementally’ tab or
4. Select the ‘Jog to position’ tab.

If an automatic program is paused axes that are not enabled outside of the ‘Manual control’ tab can be enabled and jogged. Homing is blocked as soon as an automatic program is active.

Figure 4-4: Jog modes - overview

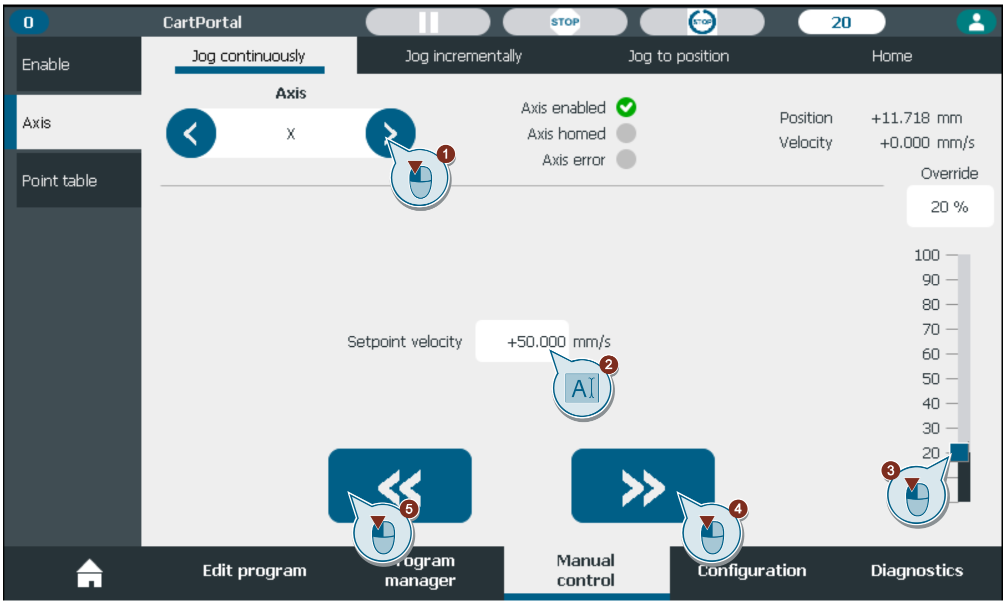


Jog continuously

The ‘Jog continuously’ tab offers jogging of a single axis with a setpoint velocity and an override in positive and negative direction. The axis must be enabled for jogging (cf. Figure 4-5).

1. Select an axis with the arrow buttons.
2. Enter a setpoint velocity.
3. Choose an override that is multiplied with the setpoint velocity.
4. Jog the selected axis forwards.
5. Jog the selected axis backwards.

Figure 4-5: Jog continuously



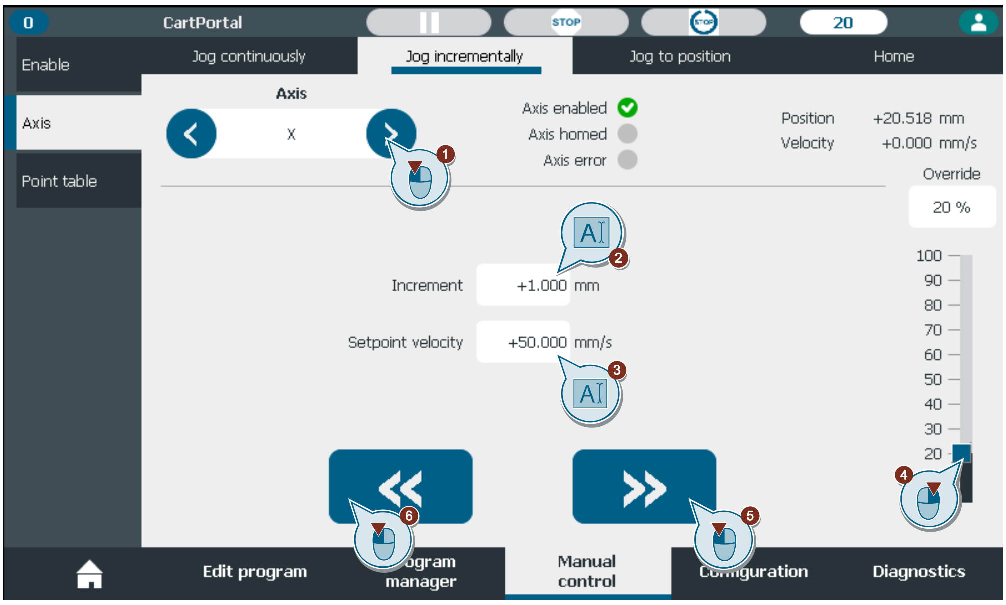
The default jog velocity can be configured in the ‘Configuration’ tab (cf. chapter 3.4.3).

Jog incrementally

The ‘Jog incrementally’ tab offers jogging of a single axis with a setpoint velocity, an increment, and an override in positive and negative direction. The axis must be enabled for jogging (cf. Figure 4-6).

1. Select an axis with the arrow buttons.
2. Enter an increment.
3. Enter a setpoint velocity.
4. Choose an override that is multiplied with the setpoint velocity.
5. Jog the selected axis forwards by the configured increment.
6. Jog the selected axis backwards by the configured increment.

Figure 4-6: Jog incrementally



Axis: X

Axis enabled ✓
Axis homed ●
Axis error ●

Position +20.518 mm
Velocity +0.000 mm/s

Override 20 %

Increment +1.000 mm

Setpoint velocity +50.000 mm/s

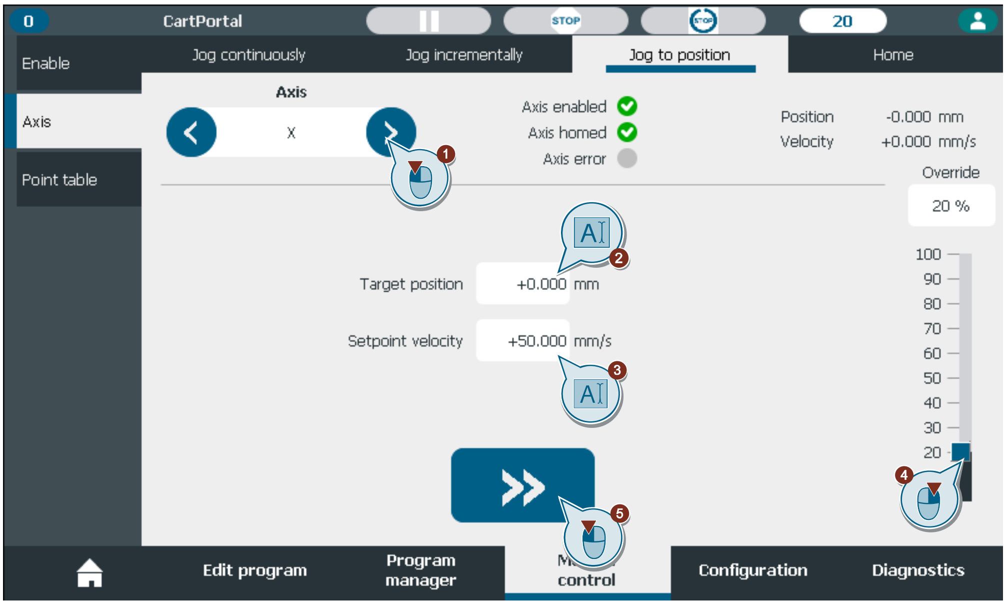
While pressing the jog button the remaining distance to reach the increment is shown. The increment is restarted when releasing the jog button. The default jog velocity and increment can be configured in the ‘Configuration’ tab (cf. chapter 3.4.3).

Jog to position

The ‘Jog to position’ tab offers jogging of a single axis with a setpoint velocity and an override to a target position. The axis must be enabled and homed for jogging (cf. Figure 4-7).

1. Select an axis with the arrow buttons.
2. Enter a target position.
3. Enter a setpoint velocity.
4. Choose an override that is multiplied with the setpoint velocity.
5. Jog the selected axis to the target position.

Figure 4-7: Jog to position



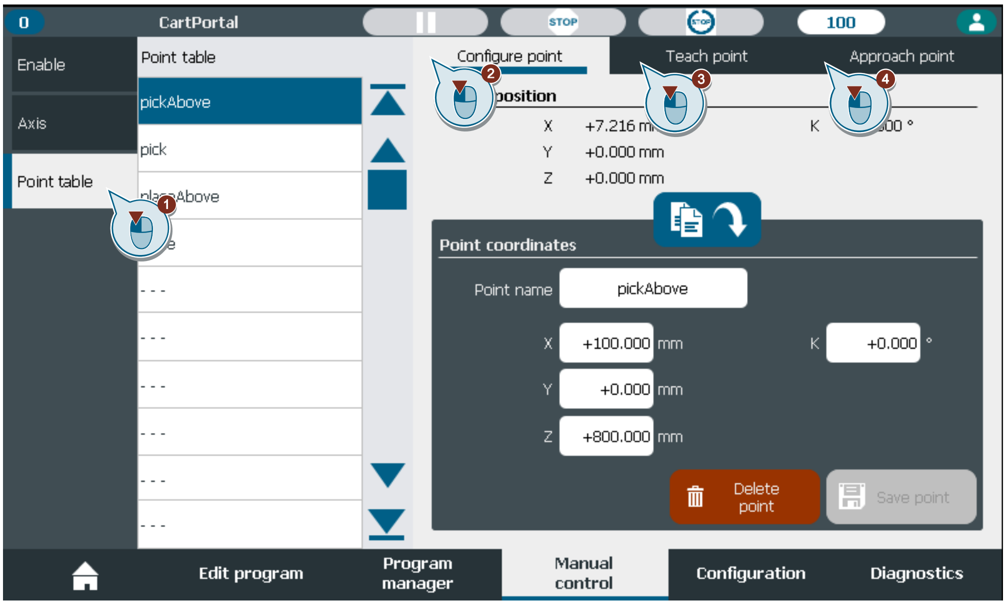
While pressing the jog button the remaining distance to reach the target position is shown. The default jog velocity can be configured in the ‘Configuration’ tab (cf. chapter 3.4.3).

4.4 Point table

SIMATIC Cartesian Portal offers three modes to interact with points in the point table. To navigation to the point table tabs (cf. Figure 4-8)

1. Select the ‘Point table’ tab.
2. Select the ‘Configure point’ tab or
3. Select the ‘Teach point’ tab or
4. Select the ‘Approach point’ tab.

Figure 4-8: Point table - overview



If the program execution is paused the ‘Configure point’ tab can be used to modify existing points. The next sPTP command that refers to this point will execute with the updated point. All other tabs are locked during automatic program execution.

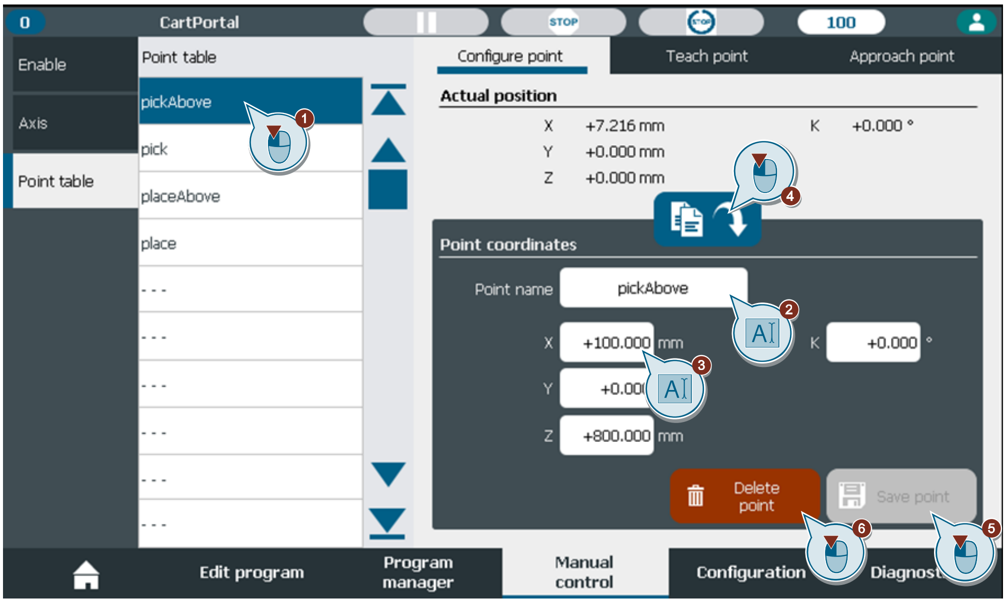
WARNING	**The override is set to 100% in the point table.** The kinematics might move faster than expected. Choose a low velocity in the ‘Teach point’ or ‘Approach point’ tab.

4.4.1 Configure point

The ‘Configure point’ tab offers the manual configuration of a point in the point table (cf. Figure 4-9).

1. Select a point to configure.
2. Enter/ change the point name.
3. Enter/ change the positions for each kinematics axis or
4. Copy the actual position of each kinematics axis.
5. Save the configured point.
6. Delete the selected point.

Figure 4-9: Manually configure a point



Point table: pickAbove, pick, placeAbove, place, - - -, - - -, - - -, - - -, - - -, - - -

Actual position: X +7.216 mm, Y +0.000 mm, Z +0.000 mm, K +0.000 °

Point coordinates: Point name pickAbove, X +100.000 mm, Y +0.00(, Z +800.000 mm, K +0.000 °

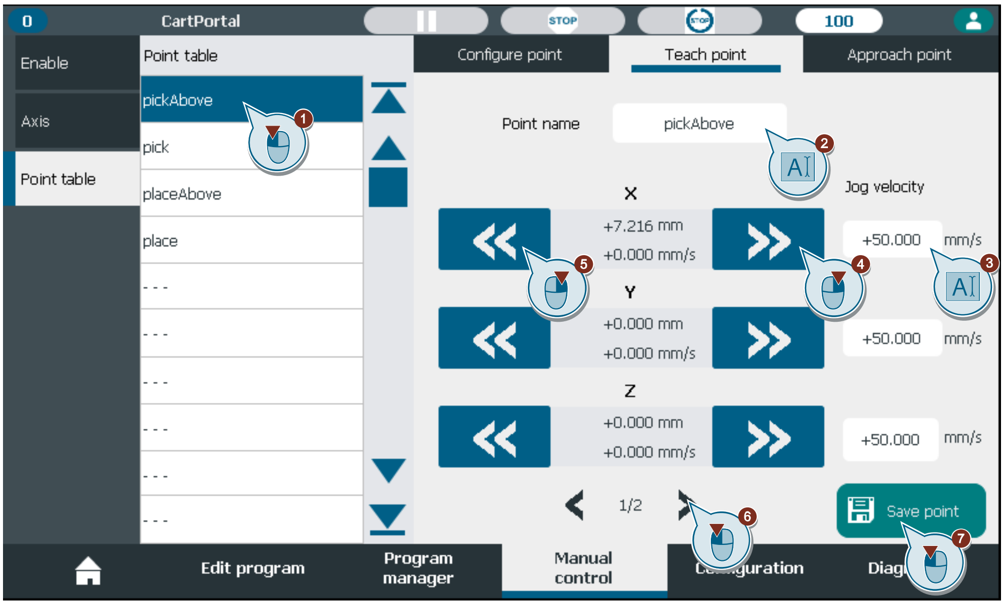
4.4.2 Teach point

The ‘Teach point’ tab offers jogging of each kinematics axis individually (cf. Figure 4-10). The resulting position is then saved in a selected point in the point table. All kinematics must be enabled, homed and no zone collision must be active.

1. Select a point to configure.
2. Enter/ change the point name.
3. Enter the jog velocity.
4. Jog the axis forwards.

5. Jog the axis backwards.
6. Switch to the second page of axes (if more than three kinematics axes are configured).
7. Save the current position and name to the selected point.

Figure 4-10:Teach a point



Point name pickAbove

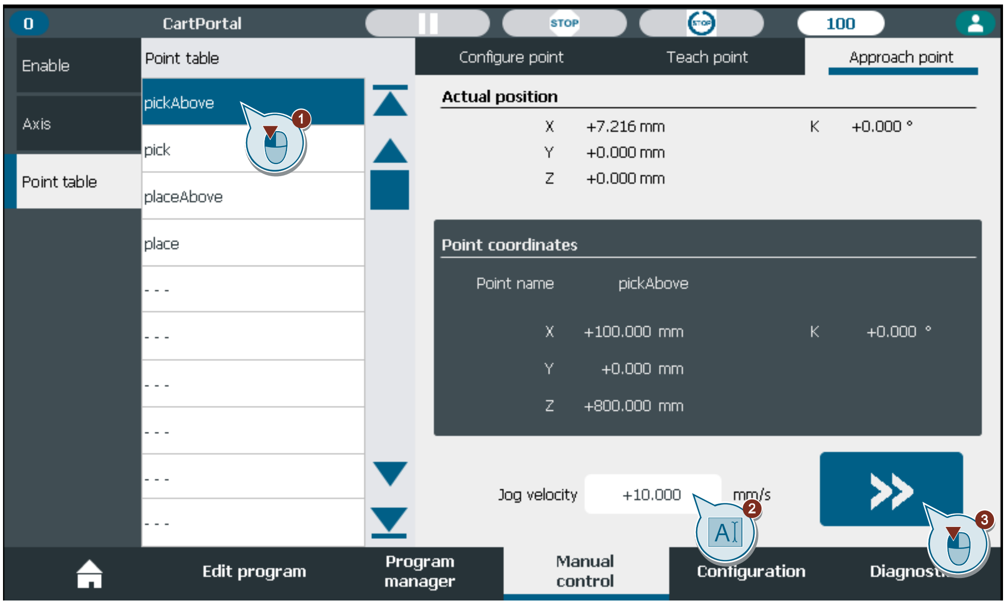
X	+7.216 mm +0.000 mm/s	+50.000	mm/s
Y	+0.000 mm +0.000 mm/s	+50.000	mm/s
Z	+0.000 mm +0.000 mm/s	+50.000	mm/s

4.4.3 Approach point

The ‘Approach point’ tab offers approaching configured points with an sPTP motion (cf. Figure 4-11). All kinematics must be enabled, homed and no zone collision must be active.

1. Select a point to approach.
2. Enter a jog velocity.
3. Approach the point position with sPTP.

Figure 4-11:Approach a point



Actual position

X +7.216 mm K +0.000 °
Y +0.000 mm
Z +0.000 mm

Point coordinates

Point name pickAbove

X +100.000 mm K +0.000 °
Y +0.000 mm
Z +800.000 mm

Jog velocity +10.000 mm/s

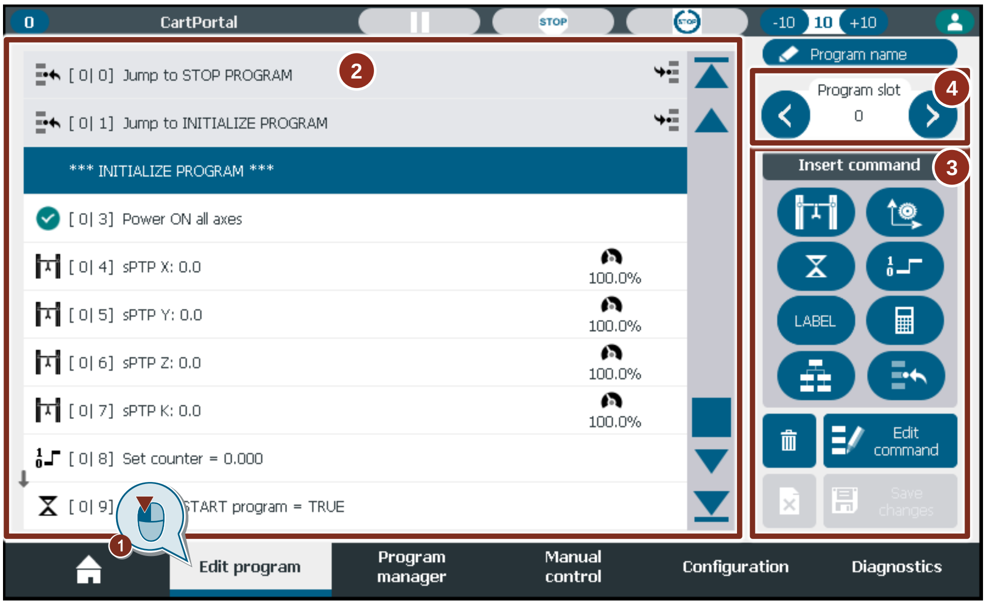
5 Programming

5.1 Overview of program editor

The program editor can be used to create a program by addition, removing or editing commands (cf. Figure 5-1). Additionally, a running program with all active sequences can be monitored (cf. chapter 7.3).

1. Select the program editor in the navigation by clicking on ‘Edit program’.
2. The command list is displayed on the left side. It displays a section of up to ten configured commands of the selected program slot. Select a command the corresponding line by clicking on it.
3. The command bar includes different commands to structure the program and move the axes. The commands are described in the following chapters.
4. The program slot can be selected by pressing the right or left arrow buttons. The active program slot is shown in the middle.

Figure 5-1: Overview program editor



[0| 0] Jump to STOP PROGRAM

[0| 1] Jump to INITIALIZE PROGRAM

*** INITIALIZE PROGRAM ***

[0| 3] Power ON all axes

[0| 4] sPTP X: 0.0 100.0%

[0| 5] sPTP Y: 0.0 100.0%

[0| 6] sPTP Z: 0.0 100.0%

[0| 7] sPTP K: 0.0 100.0%

[0| 8] Set counter = 0.000

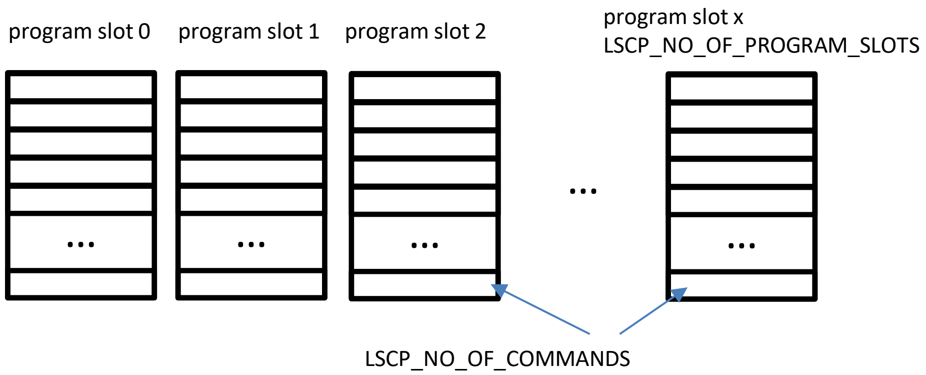
[0| 9] START program = TRUE

Program table

The program table consists of several program slots and several commands per slot (cf. Figure 5-2). In principle it is possible to program the entire automatic program in one program slot without any restrictions in the functionality of the program. However, the program slots are useful to structure the automatic program visually. So, it is recommended to use labels and jumps to structure the program.

A program is always executed sequentially with ascending command number in the respective program slot.

Figure 5-2: Program table



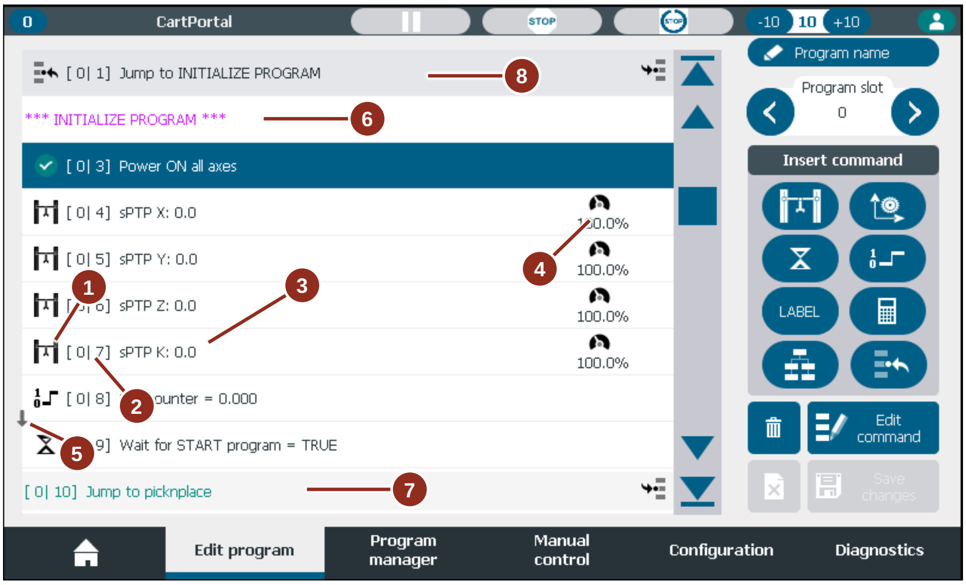
Command list

Every line describes a command that has the following structure (cf. Figure 5-3):

1. Each command has an icon.
2. Each command has a program slot number and a command number.
3. The command name with the associated parameters provides further details on the command configuration.
4. Additional icons at the end of the command can provide more information.
5. An arrow icon between the commands indicates a transition to the next command in the next or in the same application cycle.
6. Labels are marked with *** at the beginning and at the end of the name and are displayed in pink.
7. All commands with a reference to a label are displayed in green.
8. Default command that cannot be changed have a grey background.

The last available line in the program slot is marked in black color. No further commands can be added then.

Figure 5-3: Overview program editor command lines


[0| 1] Jump to INITIALIZE PROGRAM

*** INITIALIZE PROGRAM ***

[0| 3] Power ON all axes

[0| 4] sPTP X: 0.0 100.0%

[0| 5] sPTP Y: 0.0 100.0%

[0| 6] sPTP Z: 0.0 100.0%

[0| 7] sPTP K: 0.0 100.0%

[0| 8] ounter = 0.000

9] Wait for START program = TRUE

[0| 10] Jump to pickplace

Icon overview

Table 5-1 shows all icons of the program editor.

Table 5-1: Icons in command line

Icon	Description
╓┴╖	Kinematics sPTP command
⚙	Single axis command
⧗	Wait command
STOP	Stop motion command
1 0	Set output or variable
LABEL	Insert a label
↶	Jump command
↷	Button to navigate to the referenced label in the program editor
🖩	Calculate command
☷	Parallel sequence
↷	Blending is configured
◔	Velocity factor with value in %
🗑	Delete commands, program slots or the whole program
✎	Edit command
✕	Discard changes
💾	Save changes
⧉	Copy the selected command
⊕	Insert the copied command at the selected line
🔧	User-specific settings, such as blending, dynamics, etc.
↶	Return to the view before.

5.2 Basic editor functionalities

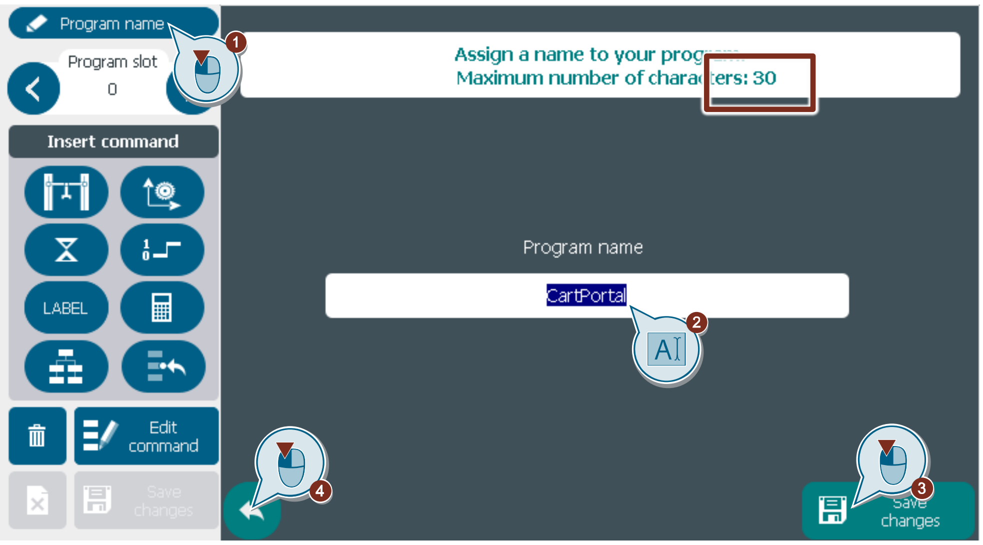
Assign a program name

To edit the program name, press the ‘program name’ button. Enter the desired name in the configuration mask and save the changes or discard the changes by using the return button (cf. Figure 5-4). The maximum number of characters that can be used for the name is displayed.

NOTE

The program name is also the channel name of the current channel.

Figure 5-4: Assign program name



Assign a name to your program
Maximum number of characters: 30

Program name

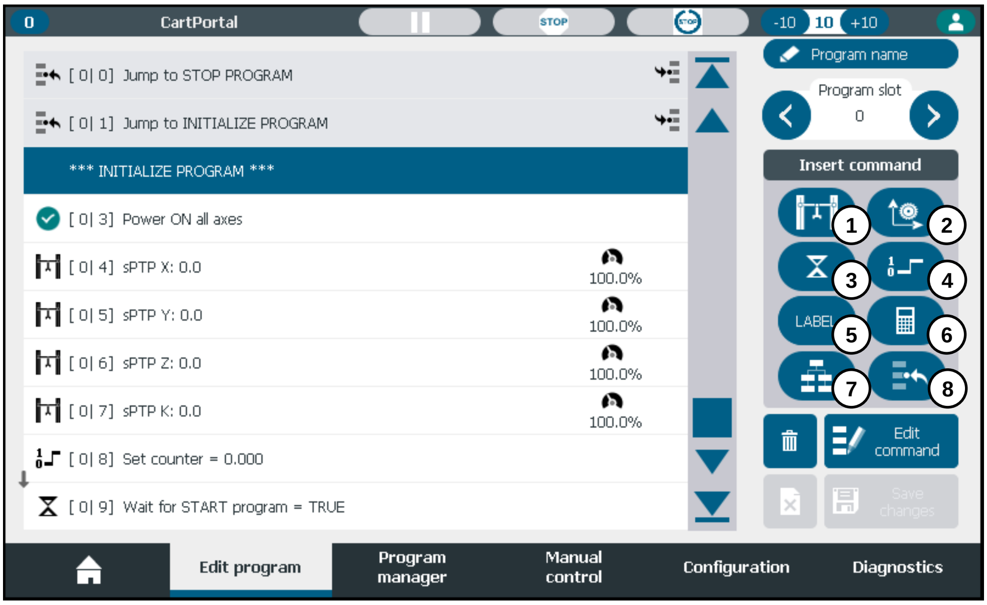
CartPortal

Insert a command

To insert a command, select a command line in a program slot and then select the command that should be added in the command bar. When the command is saved it is added above the selected command line and all subsequent commands are shifted down by one line. The initially selected command stays selected. The program editor offers the following commands (cf. Figure 5-5):

1. Kinematics commands (cf. chapter 5.3).
2. Single axis commands (cf. chapter 5.4).
3. Wait commands (cf. chapter 5.5).
4. Set commands (cf. chapter 5.6).
5. Label command (cf. chapter 5.7).
6. Calculator command (cf. chapter 5.8).
7. Parallel sequence commands (cf. chapter 5.9).
8. Jump/Loop commands (cf. chapter 5.10).

Figure 5-5: Add a command



[0| 0] Jump to STOP PROGRAM
[0| 1] Jump to INITIALIZE PROGRAM
*** INITIALIZE PROGRAM ***
[0| 3] Power ON all axes
[0| 4] sPTP X: 0.0 100.0%
[0| 5] sPTP Y: 0.0 100.0%
[0| 6] sPTP Z: 0.0 100.0%
[0| 7] sPTP K: 0.0 100.0%
[0| 8] Set counter = 0.000
[0| 9] Wait for START program = TRUE

Edit a command

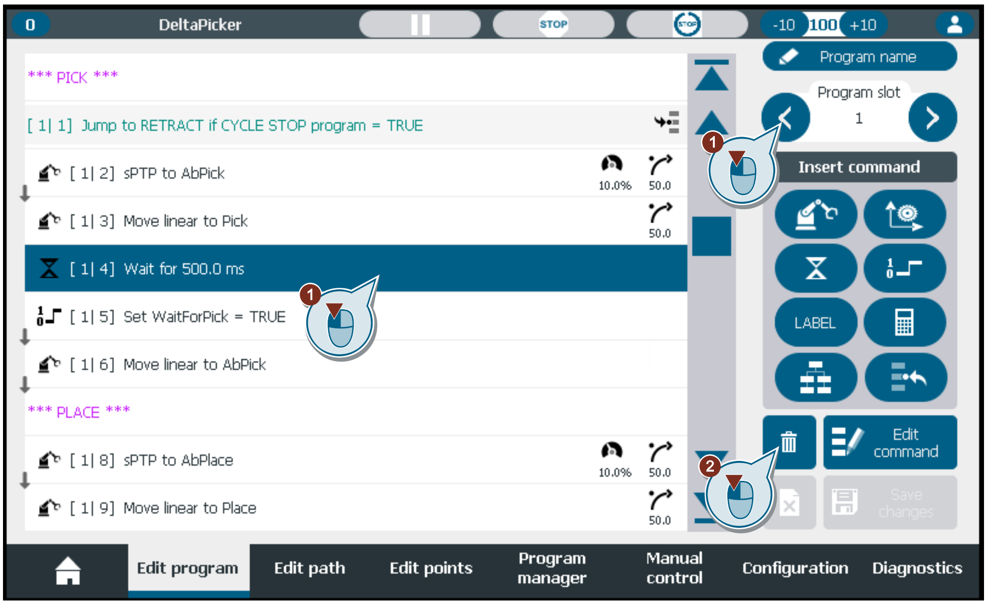
To edit a command, select it in the table and then press the ‘Edit command’ button. Depending on the selected command, the corresponding configuration mask opens, and the parameters can be adjusted.

Delete commands, program slot or the complete program

It is possible to delete either single command lines, whole program slots or the complete program. The procedure for this is as follows (cf. Figure 5-6 and Figure 5-7):

1. Select the command that should be deleted.
2. Click the delete button in the command bar.

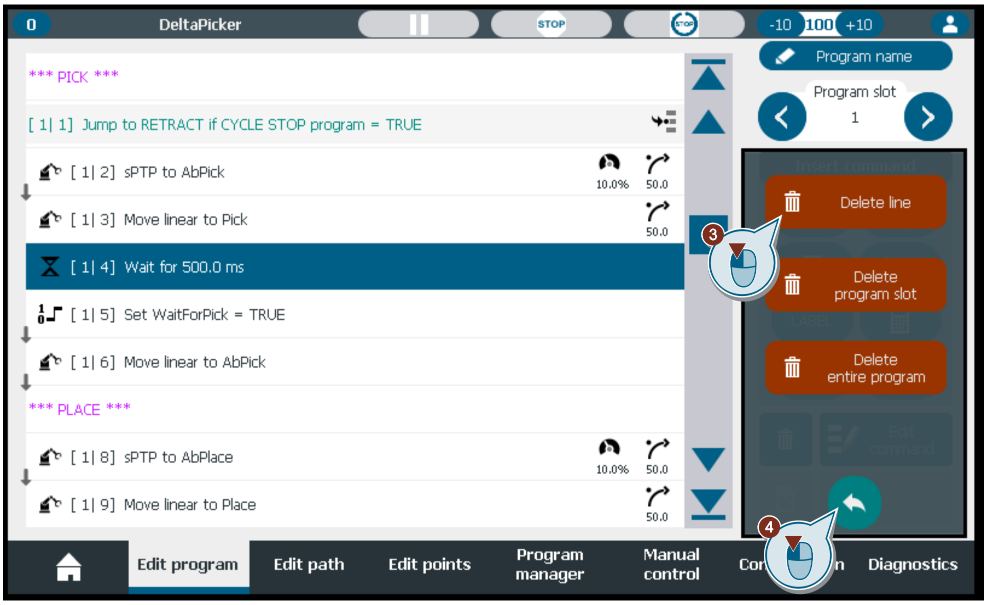
Figure 5-6: Delete command with command bar



*** PICK ***
[1| 1] Jump to RETRACT if CYCLE STOP program = TRUE
[1| 2] sPTP to AbPick 10.0% 50.0
[1| 3] Move linear to Pick 50.0
[1| 4] Wait for 500.0 ms
[1| 5] Set WaitForPick = TRUE
[1| 6] Move linear to AbPick
*** PLACE ***
[1| 8] sPTP to AbPlace 10.0% 50.0
[1| 9] Move linear to Place 50.0

3. Select the data you want to delete → Line / Slot / Entire program.
4. Close the delete window by pressing the return button.

Figure 5-7: Selection delete

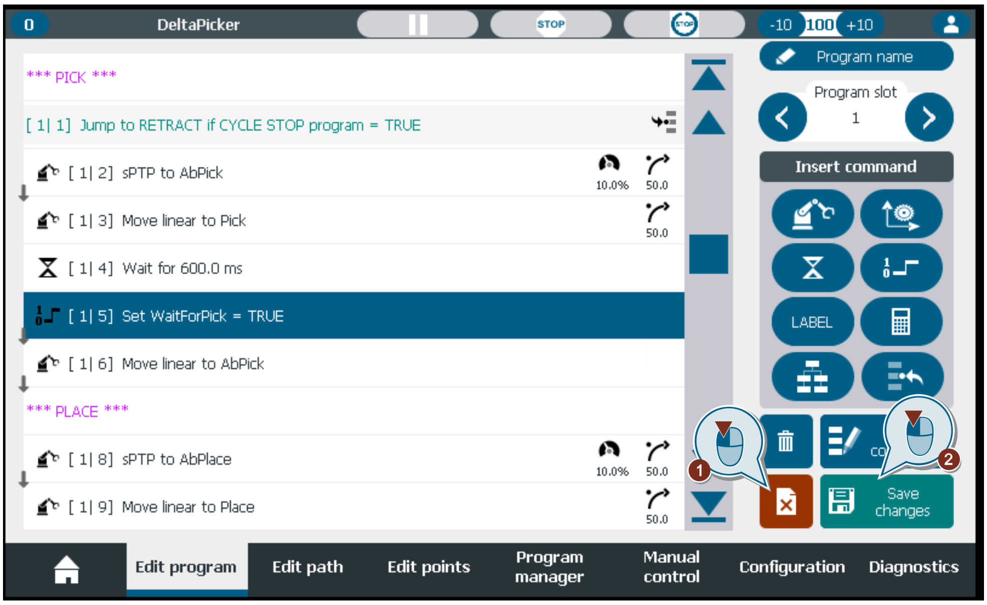


*** PICK ***
[1| 1] Jump to RETRACT if CYCLE STOP program = TRUE
[1| 2] sPTP to AbPick 10.0% 50.0
[1| 3] Move linear to Pick 50.0
[1| 4] Wait for 500.0 ms
[1| 5] Set WaitForPick = TRUE
[1| 6] Move linear to AbPick
*** PLACE ***
[1| 8] sPTP to AbPlace 10.0% 50.0
[1| 9] Move linear to Place 50.0

Save or discard changes on the program

All changes to the program are detected and the green 'Save changes' button (2) is activated (cf. Figure 5-8: Save changes). Program changes are retained when the user switches to other screens, to create new variables, points, or path segments, for example. However, changes are discarded when the 'Program manager' or the 'Home screen' is called. By pressing the button, all changes will be finally applied.

Figure 5-8: Save changes



*** PICK ***
[1| 1] Jump to RETRACT if CYCLE STOP program = TRUE
[1| 2] sPTP to AbPick 10.0% 50.0
[1| 3] Move linear to Pick 50.0
[1| 4] Wait for 600.0 ms
[1| 5] Set WaitForPick = TRUE
[1| 6] Move linear to AbPick
*** PLACE ***
[1| 8] sPTP to AbPlace 10.0% 50.0
[1| 9] Move linear to Place 50.0

If you have made changes to the program that you do not want to apply, press the 'Discard' button (1). All unsaved changes will then be discarded without warning.

NOTICE	**Unsaved program changes will be discarded!** Make sure that you save all desired changes before you switch to the Home or Program manager screen, otherwise all changes will be discarded.

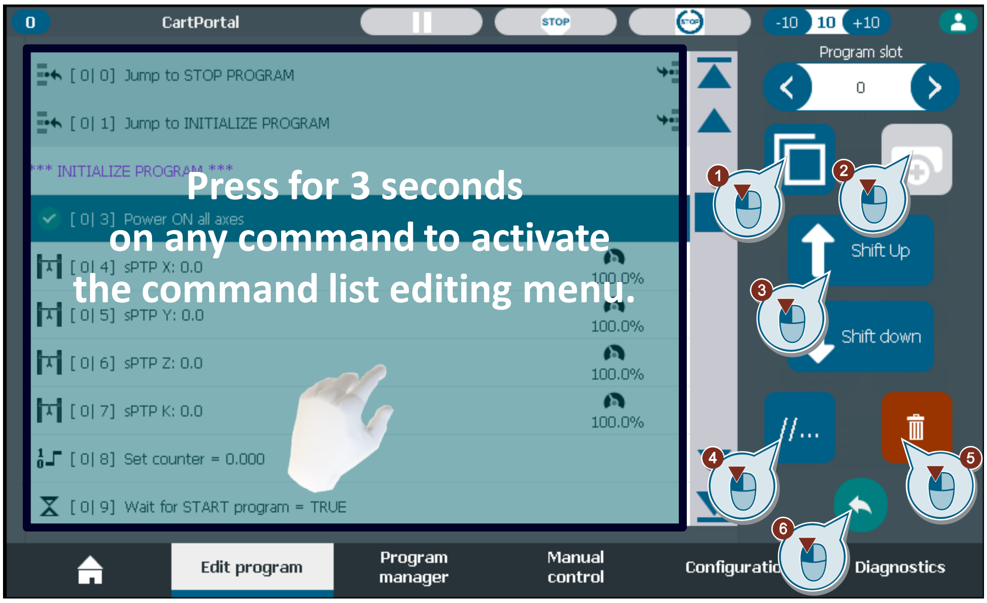
Copy & paste commands

To modify the command list, the user can access additional auxiliary functions, such as copy and paste, delete, or comment out a command. These functions are grouped in a separate submenu and are independent of the type of commands. To open this editing menu, any line at the program list must be pressed for at least 3 seconds.

The menu offers the following functions (cf. Figure 5-9):

1. Copy the selected command.
2. Paste from the previously copied command.
3. Shift the selected command up or down within the program slot.
4. Deactivate the selected command by commenting it out or reactivate the previously commented out command. A commented-out command will be ignored during execution.
5. Delete the selected command.
6. Exit the submenu.

Figure 5-9: Command list subeditor



[0| 0] Jump to STOP PROGRAM

[0| 1] Jump to INITIALIZE PROGRAM

*** INITIALIZE PROGRAM ***

[0| 3] Power ON all axes

[0| 4] sPTP X: 0.0 100.0%

[0| 5] sPTP Y: 0.0 100.0%

[0| 6] sPTP Z: 0.0 100.0%

[0| 7] sPTP K: 0.0 100.0%

[0| 8] Set counter = 0.000

[0| 9] Wait for START program = TRUE

Press for 3 seconds on any command to activate the command list editing menu.

NOTE Commands with references cannot be commented out.

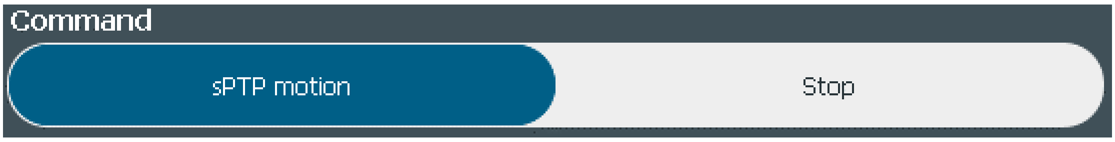
5.3 Kinematics commands

5.3.1 Overview

The SIMATIC Cartesian Portal offers the kinematics commands (cf. Figure 5-10))

- Synchronous point to point motion (cf. chapter 5.3.2)
- Stop kinematics (cf. chapter 5.3.3)

Figure 5-10: Kinematics commands



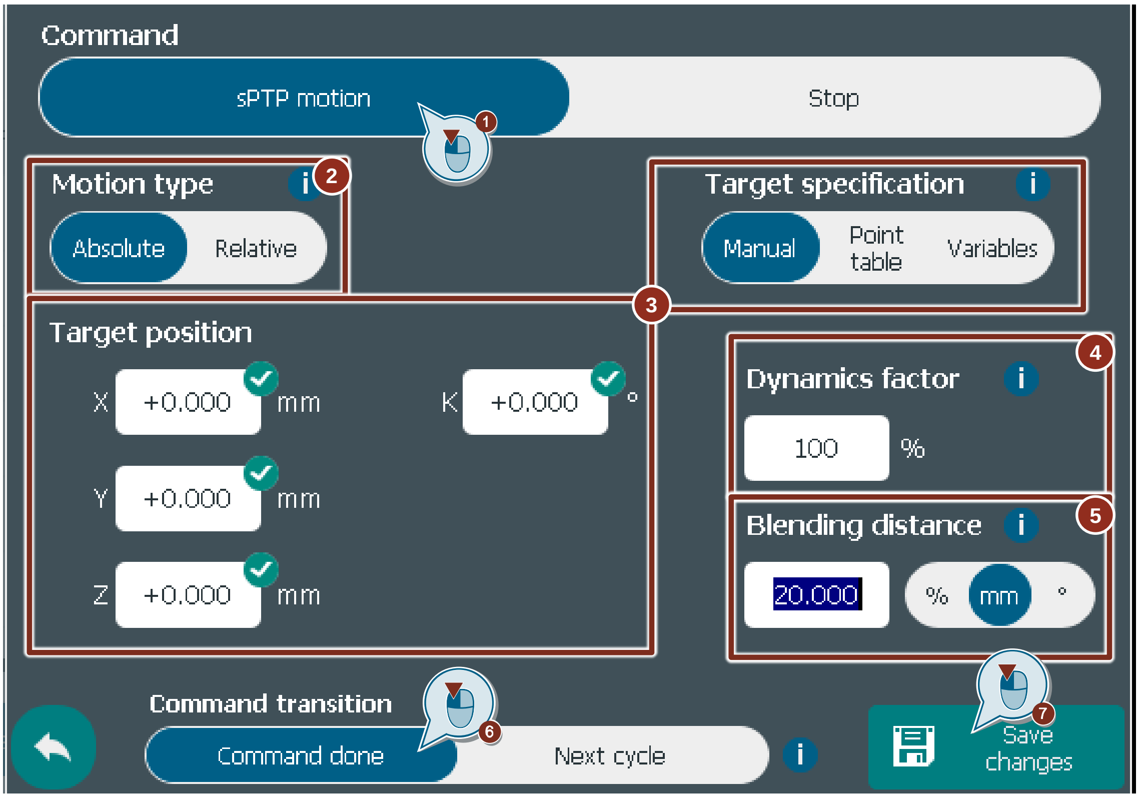
5.3.2 Synchronous point to point motion

The sPTP command enables the synchronized positioning of all or a subset of the kinematics axes. This means that all selected axes start and stop their motion at the simultaneously. The command is configured by (cf. Figure 5-11)

1. Select ‘sPTP’ command.
2. Select a motion type
   - Absolute: All axes will be position to the target position.
   - Relative: The target position is added to the target position of the currently active sPTP command or to the current position of each axis if no sPTP command is active.
3. Select a target specification, the target value and select which kinematics axes are synchronized.
4. Enter a dynamics factor.

5. Select a blending mode and enter a blending distance.
6. Select a command transition.
7. Save the changes.

Figure 5-11: Kinematics command – Synchronous point to point motion



Target specification – Manual

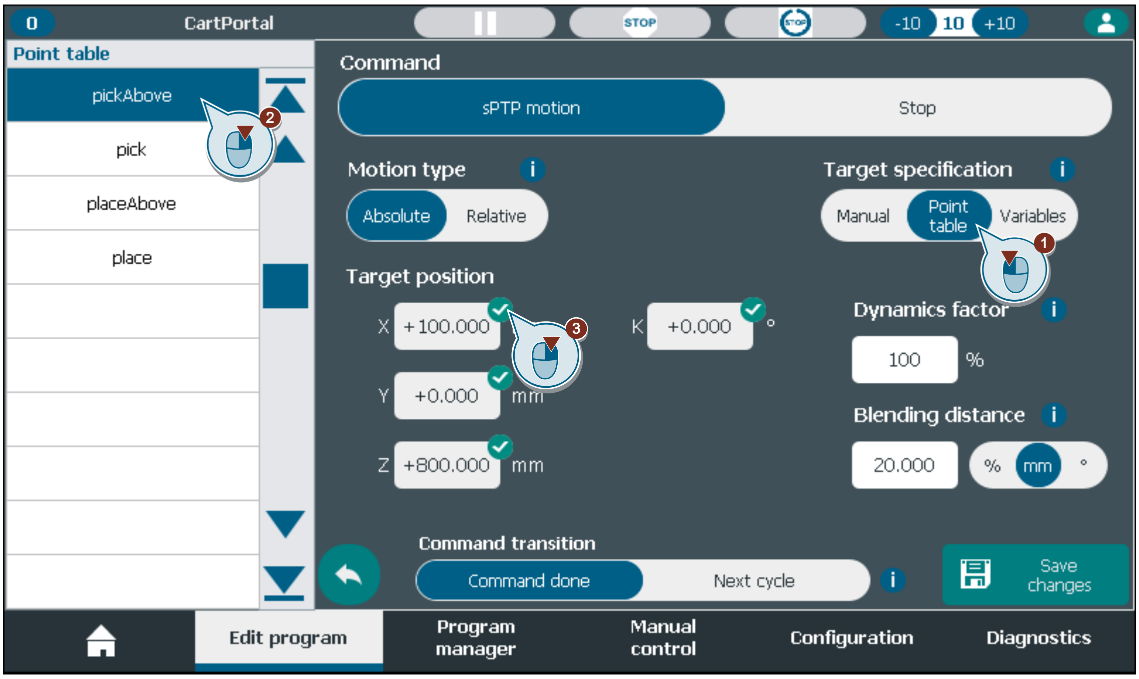
With the target specification ‘Manual’ a fixed value can be entered for each axis. As soon as a value is entered the axis will automatically be selected for the synchronization of that sPTP command. An axis can be deselected afterwards by clicking on the checkmark in the top right corner of the respective axis target position.

Target specification – Point table

With the target specification ‘Point table’ a point that was previously saved in the ‘Manual control’ tab (cf. chapter 4.4) can be referenced as the target position. This reference is evaluated in the moment the command is executed. Changes to the referenced point will thus be active immediately, if the command is not currently executing. To configure this command (cf. Figure 5-12)

1. Select the ‘Point table’ as the ‘Target specification’. This opens the point table on the left.
2. Select a point from the point table (only configured points are shown).
3. (Optional) Deselect an axis from the synchronization.

Figure 5-12: Synchronous point to point motion – Point table reference



X +100.000

Y +0.000 mm

Z +800.000 mm

K +0.000 °

Dynamics factor 100 %

Blending distance 20.000 mm

NOTICE

Deselect an axis with a point table reference

Points are always configured for all kinematics axes.

Ensure that the selected point is also valid for the selected subset of kinematics axes.

Target specification – Variable in combination with manual or point table

With the target specification ‘Variable’ a variable that was previously configured in the ‘Configuration’ tab (cf. chapter 3.7.5) can be referenced as the target position of an axis. This reference is evaluated in the moment the command is executed. Changes to the referenced variable will thus be active immediately, if the command is not currently executing. Each axis target position can reference a separate variable. The target positions of axes that have no referenced variable can be specified manually, by a point from the point table or they can be deselected. To configure this command (cf. Figure 5-13)

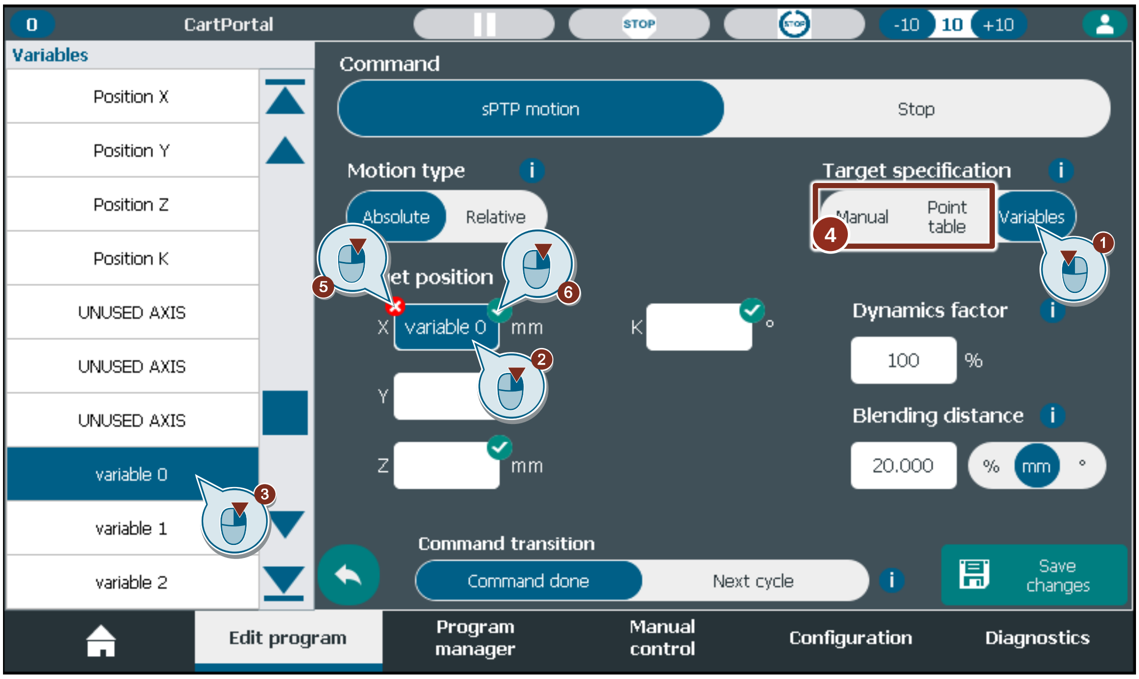
1. Select ‘Variables’ as ‘Target specification’.
2. Select the axis target position to reference a variable.
3. Select the variable to reference.
4. (Optional) Switch to ‘Manual’ or ‘Point table’ to specify all other axis target positions.
5. (Optional) Delete the variable reference.
6. (Optional) Deselect the axis from synchronization.

NOTICE

Variable reference is more dominant than point reference!

If a combination of variables and point reference is desired, first assign variables to the corresponding target coordinates and then select a point from the point table. The coordinates assigned with variables overwrite the point coordinates of the point reference.

Figure 5-13: Synchronous point to point motion – Variable reference



Blending

To achieve an uninterrupted kinematics motion blending between motion commands can be applied. The individual motion commands are then blended depending on the remaining distance of the currently active command. Therefore, the blending value of a command determines when this command enters the blending segment and can transition into the next sPTP command.

The blending consists of a blending mode (1) and a blending value (2). The blending modes are described in Table 5-2. The blending value is defined by

- Value = 0: deactivates blending. The Kinematics movement is stopped at the target position of the command.
- Value > 0: directly defines the blending. The unit depends on the selected mode.
  If the value exceeds the maximum blending distance, it is limited to the maximum value (90%).
- Value < 0 defined the blending to the maximum value. Depending on the blending mode the maximum value is
  - mm: 90% of the shortest distance to the target of all linear kinematics axes involved in the sPTP command (cf. Figure 5-15).
  - °: 90% of the shortest distance to the target of all rotary kinematics axes involved in the sPTP command.
  - %: 90% of the remaining distance of all kinematics axes involved in the sPTP command.

Figure 5-14: Synchronous point to point motion – Blending

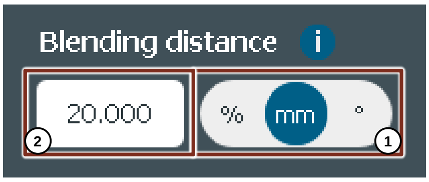
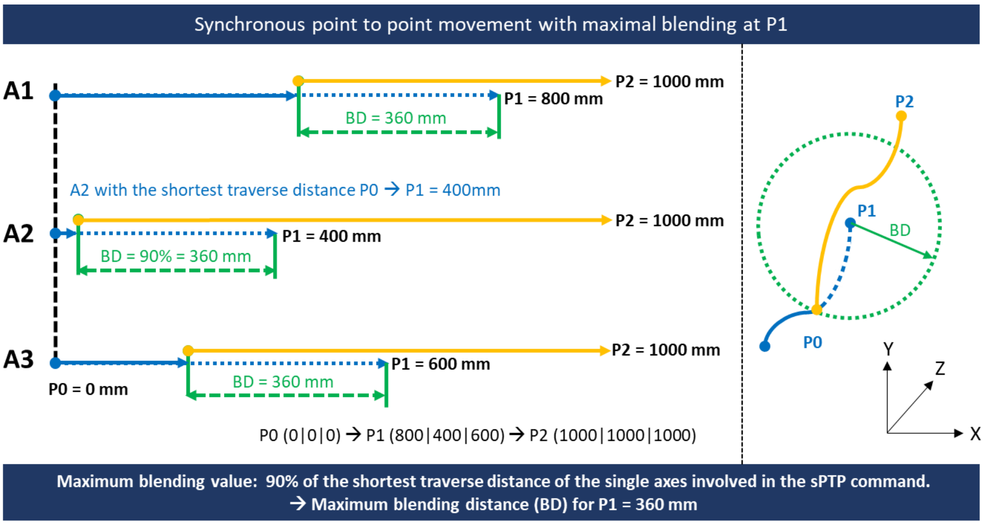


Table 5-2: Synchronous point to point motion – Blending modes

Mode	Description
mm	Only linear axes of the Kinematics are considered for blending. A sPTP command is blended by the next sPTP command when the remaining distance to the target position of all linear axes is below the blending distance in mm.
°	Only rotatory axes of the Kinematics are considered for blending. A sPTP command is blended by the next sPTP command when the remaining distance to the target position of all rotatory axes is below the blending distance in °.
%	All axes of the Kinematics are considered for blending. A sPTP command is be blended by the next sPTP command when the remaining distances of all axes is below the blending distance to their target positions in %.

Figure 5-15: Synchronous point to point motion - Maximum blending distance


Synchronous point to point movement with maximal blending at P1
A1
P2 = 1000 mm
P1 = 800 mm
BD = 360 mm
A2 with the shortest traverse distance P0 → P1 = 400mm
A2
P2 = 1000 mm
P1 = 400 mm
BD = 90% = 360 mm
A3
P2 = 1000 mm
P1 = 600 mm
BD = 360 mm
P0 = 0 mm
P0 (0|0|0) → P1 (800|400|600) → P2 (1000|1000|1000)
P2
P1
BD
P0
Y
Z
X
Maximum blending value: 90% of the shortest traverse distance of the single axes involved in the sPTP command.
→ Maximum blending distance (BD) for P1 = 360 mm

Dynamics

The dynamics factor scales the velocity, acceleration, deceleration, and jerk of the configured dynamics limits of each axis of the kinematics. The target velocity of each axis is then scaled further with the override and again such that all axes are synchronized.

Transition

The sPTP command offers the command transition “Command done” and “Next cycle”. The command transition “Command done” proceeds with the next command when either the target position is reached or if configured, the blending segment is reached.

The command transition “Next cycle” proceeds with the next command in the next PLC cycle. The next sPTP command is only executed if the last sPTP command reached its target position or the configured blending segment. Therefore, a sPTP command cannot override a running sPTP command.

Display in the program table

Depending on the configuration of the sPTP command the command is displayed differently in the program table. The command always shows all defined positions when selected (cf. Figure 5-16).

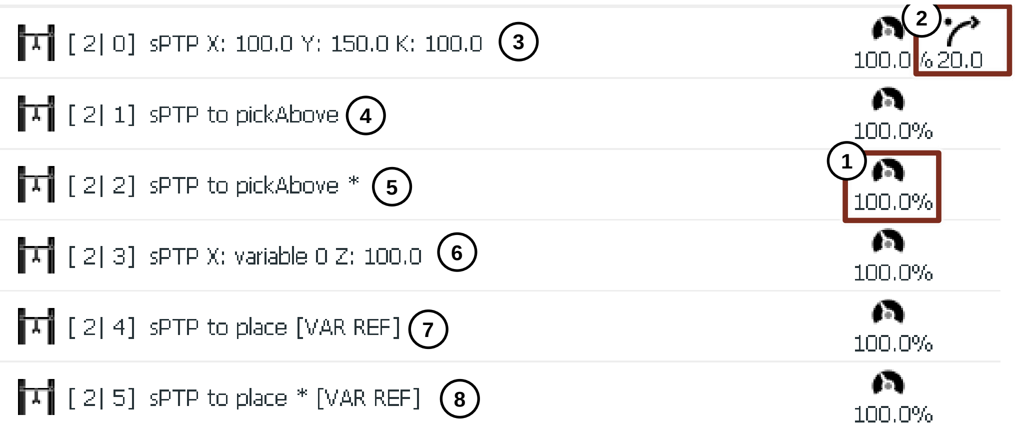
1. Shows the dynamics factor of the sPTP command.
2. Shows the blending value. The blending mode is not shown. This is only visible if a blending is configured.
3. sPTP command where X, Y and K are defined manually.
4. sPTP command to a point from the point table.

5. sPTP command to a point from the point table with a subset of the kinematics axes.
6. sPTP command where X is referenced by a variable and Z is defined manually.
7. sPTP command to a point where at least one position is referenced by a variable.
1. sPTP command to a point where at least one position is reference and only a subset of kinematics axes is used.

NOTE

If a target position of an sPTP command is the same as the target command of the previous sPTP command in the value is not shown for a cleaner appearance.

Figure 5-16: Synchronous point to point motion –Display in the program table



[2| 0] sPTP X: 100.0 Y: 150.0 K: 100.0 100.0 % 20.0

[2| 1] sPTP to pickAbove 100.0%

[2| 2] sPTP to pickAbove * 100.0%

[2| 3] sPTP X: variable 0 Z: 100.0 100.0%

[2| 4] sPTP to place [VAR REF] 100.0%

[2| 5] sPTP to place * [VAR REF] 100.0%

5.3.3 Stop command

The ‘Stop’ command stops all kinematics axes with a defined deceleration. Configure the command by (cf. Figure 5-17)

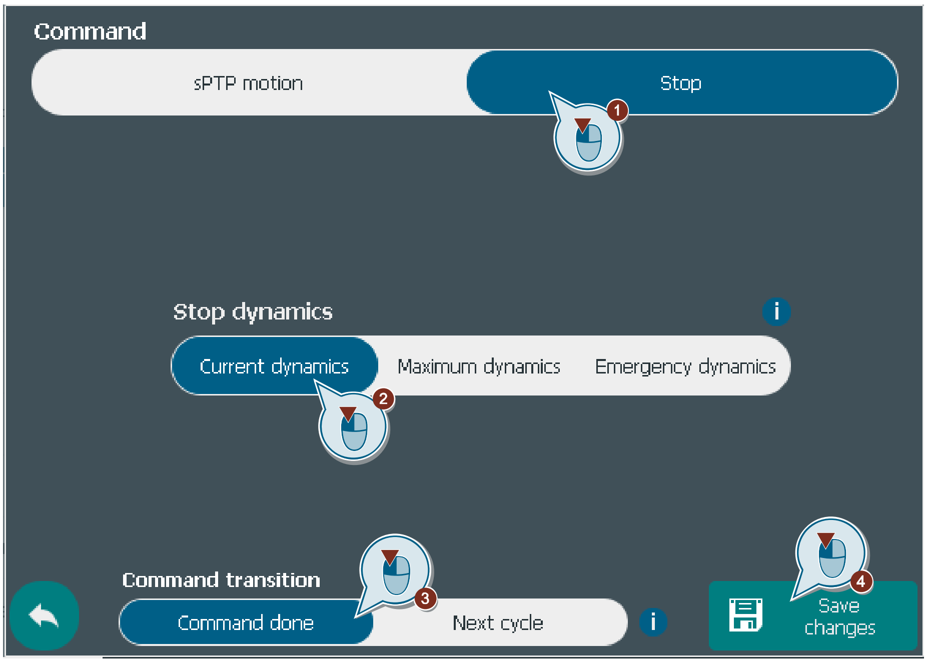
1. Select the ‘Stop’ command
2. Select the stop dynamics
   - ‘Current dynamics’: Stops all axes with the dynamics currently active at the sPTP command. If a kinematics axis moves but is not part of a sPTP command the axis is decelerated with its dynamic limits.
   - ‘Maximum dynamics’: Stops each axis with its configured dynamics limits. This mode is indicated with ‘[max dynamics]’ in the program table.
   - ‘Emergency dynamics’: Stops each axis with its configured emergency dynamics. This mode is indicated with ‘[emergency dynamics]’ in the program table.
3. Select the command transition.
4. Save the changes.

NOTICE

Kinematics axis are moving with a single axis command.

All kinematics axes are stopped. They do not have to be in motion by a sPTP command.

Figure 5-17: Stop command



5.4 Single axis commands

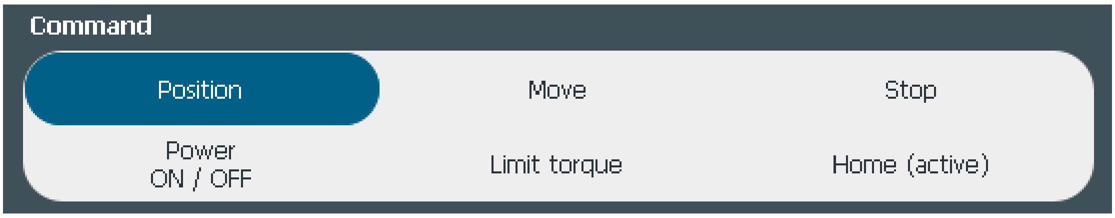
5.4.1 Overview

The SIMATIC Cartesian Portal solution offers a range of commands for controlling individual axes of type TO_PositioningAxis. Kinematics axes can be controlled both in the axis group with sPTP commands and by single axis commands.

The following single axis commands are available (cf. Figure 5-18):

- Absolute / Relative positioning (cf. chapter 5.4.2).
  (MC_MoveAbsolute / MC_MoveRelative)
- Move axis by velocity (cf. chapter 5.4.3)
  (MC_MoveVelocity)
- Stop axes with selectable dynamics (cf. chapter 5.4.4)
  (MC_Halt)
- Power ON / OFF axis (cf. chapter 5.4.5)
  (MC_Power)
- Limit axis torque (cf. chapter 5.4.6)
  (MC_TorqueLimiting)
- Active homing of axes with incremental encoder (cf. chapter 5.4.7)
  (MC_Home)

Figure 5-18: Single axis commands

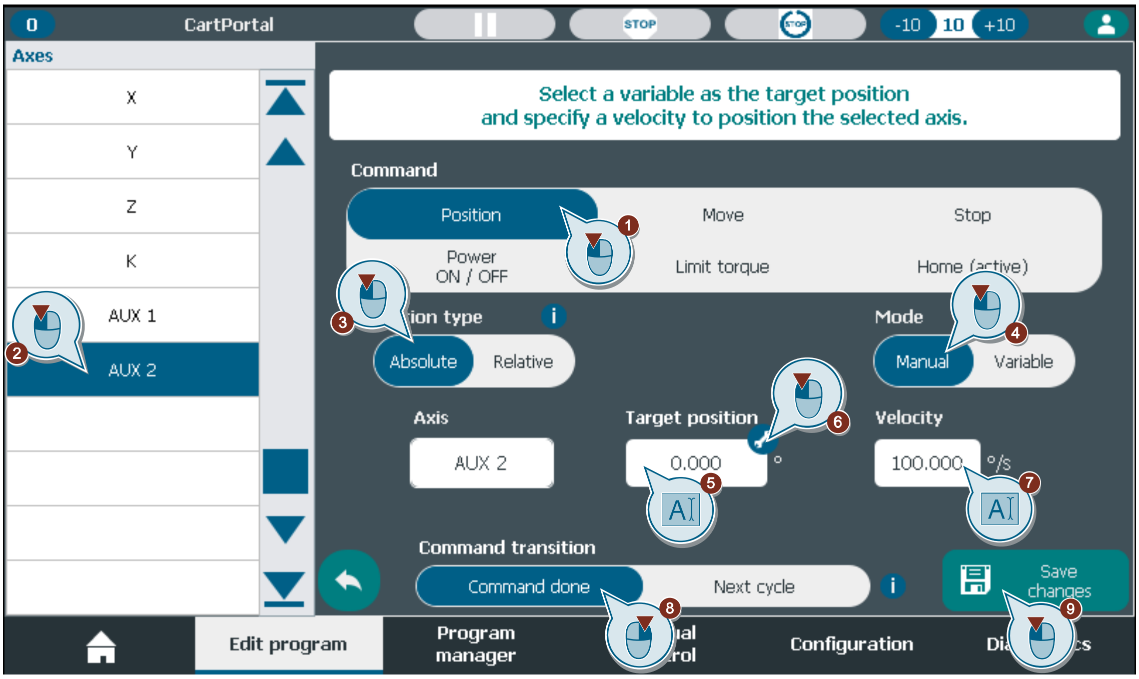


5.4.2 Absolute or relative positioning

All used axes can be positioning with the single axis positioning command. Configure the command by (cf. Figure 5-19)

1. Select the ‘Position’ command.
2. Select the axis to position (click in this field to return to the axis selection after the variable selection).
3. Select the motion type ‘Absolute’ or ‘Relative’. For relative commands the sign of the distance parameter defines the direction of the motion.
4. Select the mode ‘Manual’ to enter a fixed value or ‘Variable’ to couple the position to a variable of the SCP.

Figure 5-19: Single axis command - Position



Select a variable as the target position and specify a velocity to position the selected axis.

Axis: AUX 2

Target position: 0.000 °

Velocity: 100.000 °/s

5. Enter the target position (‘Manual’ mode) or click on this field to select a variable (‘Variable’ mode).
6. Select the ‘Direction’ setting for axis with modulo enabled. Here ‘Positive’, ‘Negative’ or ‘Shortest way’ can be selected. This button is only available for axis with enabled modulo.
7. Enter the target velocity to reach the position.
8. Select the command transition.
9. Save the command.

The dynamic defaults for acceleration, deceleration and jerk of each axis are used in addition to the configured velocity.

NOTE

> It is recommended to position a kinematics axis with the sPTP command (cf. chapter 5.3.2) by deselecting all other kinematics axes.

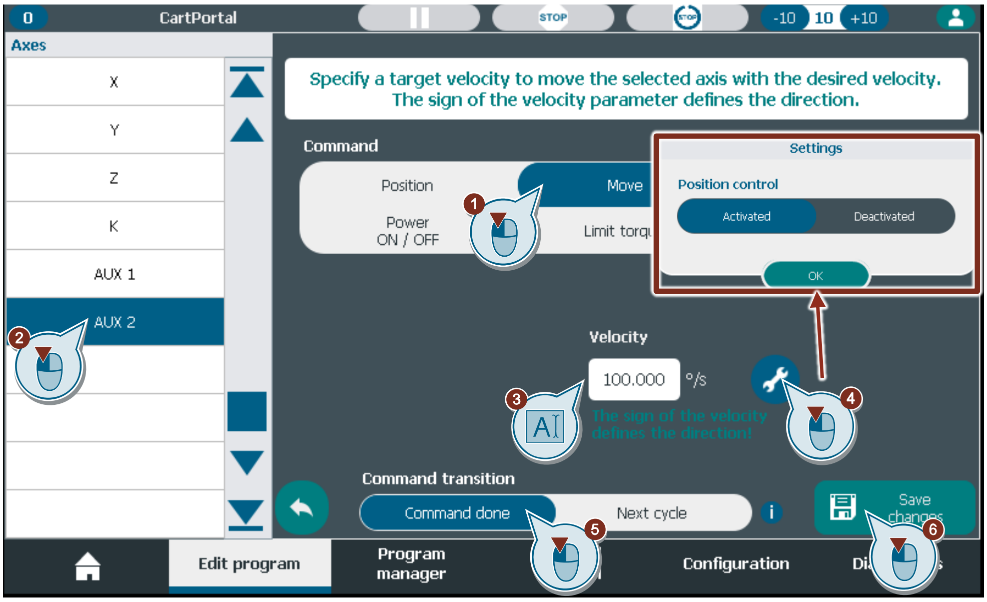
5.4.3 Move axis by velocity

All used axes can be moved with a fixed constant velocity with the ‘Move’ command. Configure the command by (cf. Figure 5-20)

1. Select the ‘Move’ command.
2. Select the axis to move.
3. Enter the desired target velocity. Note that the sign of the velocity defines the direction.
4. Activate or deactivate the position control setting. By default, the position control is activated.
5. Select the command transition. Note that a move command is done when the target velocity is reached.
6. Save the command.

The dynamic defaults for acceleration, deceleration and jerk of each axis are used in addition to the configured velocity.

Figure 5-20: Single axis command - Move



Specify a target velocity to move the selected axis with the desired velocity. The sign of the velocity parameter defines the direction.

Settings

Position control: Activated | Deactivated

Velocity: 100.000 °/s

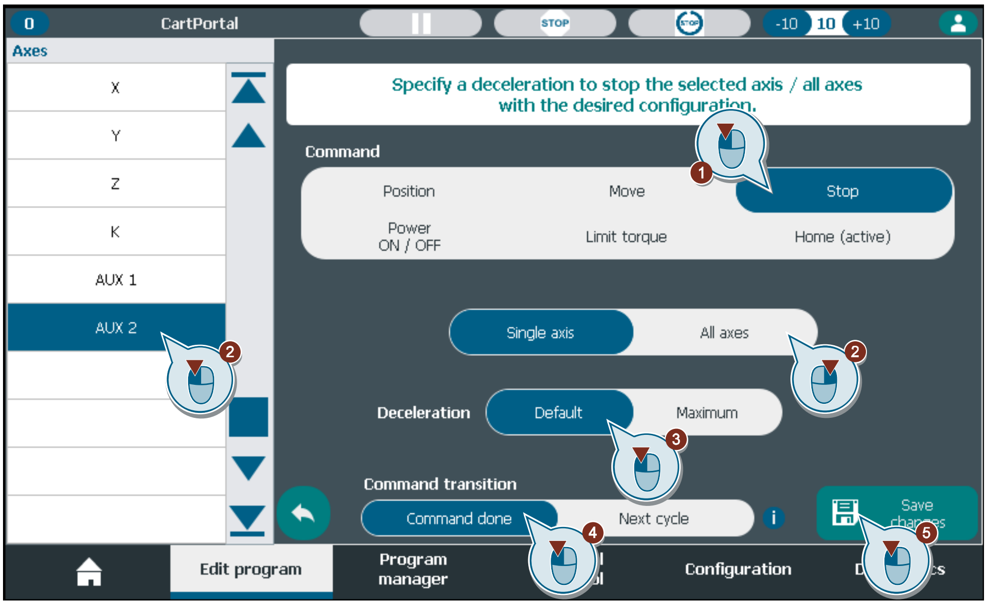
The sign of the velocity defines the direction!

5.4.4 Stop

All used axes can be stopped with a defined deceleration with the ‘Stop’ command. Configure the command by (cf. Figure 5-21)

1. Select the ‘Stop’ command.
2. Select the axis to stop or stop all axes.
3. Select the deceleration that refers to the TO defaults or TO limits of each axis.
4. Select the command transition.
5. Save the command.

Figure 5-21: Single axis command - Stop



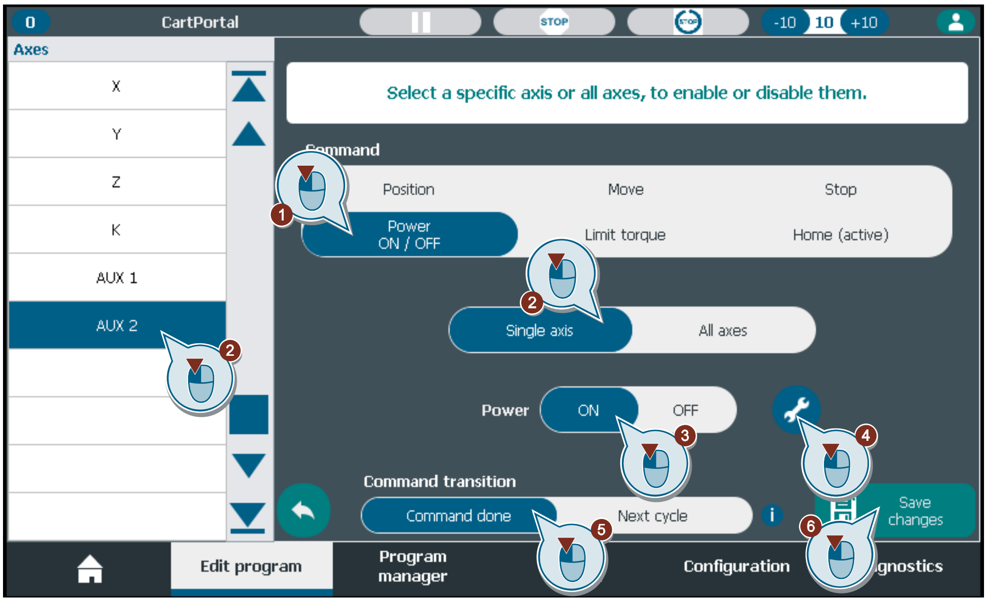
Specify a deceleration to stop the selected axis / all axes with the desired configuration.

5.4.5 Power ON / OFF

All used axes can be enabled or disabled with the ‘Power ON / Off’ command. Configure the command by (cf. Figure 5-22)

1. Select the ‘Power ON / Off’ command.
2. Select a single axis or all axes.
3. Select whether to enable or disable the selected axis/axes.
4. Configure the additional settings.
   For the ‘Power ON’ command the position control can be activated or deactivated.
   For the ‘Power Off’ command the stop mode can be specified with ‘Emergency stop’, ‘Immediate stop’, or ‘Maximum dynamics’.
5. Select the command transition.
6. Save the command.

Figure 5-22: Single axis command - Power ON / OFF



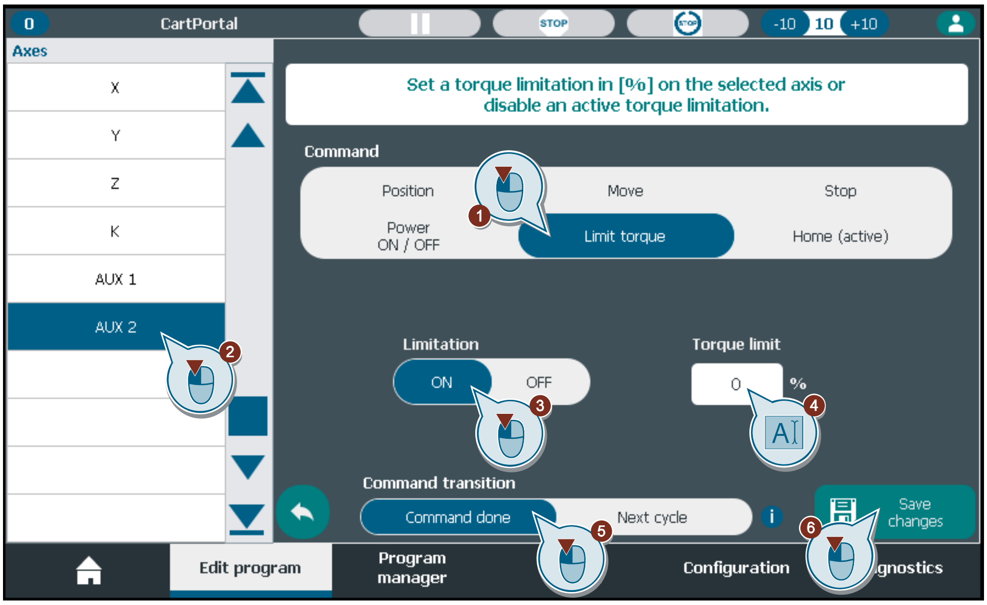
Select a specific axis or all axes, to enable or disable them.

5.4.6 Torque/Force limitation

All used axes can enable or disable torque/force limitation with the ‘Limit torque’ command. Configure the command by (cf. Figure 5-23)

1. Select the ‘Limit torque’ command.
2. Select the axis to enable or disable the torque/force limit.
3. Enter the torque/force limit in percent of the configured value in the axis limit configuration (cf. chapter 3.4.2).
4. Select the command transition.
5. Save the command.

Figure 5-23: Single axis command - Limit torque



Set a torque limitation in [%] on the selected axis or disable an active torque limitation.

NOTICE

Change the torque/force limit in the axis limit configuration.

An absolute value for the torque/force limit is saved in the command not the percentage value. This can lead to unexpected torque/force values at the axis.

Check and adapt all torque limitation commands if you have changed the torque/force value in the axis limit configuration.

5.4.7 Active homing

All used axes with incremental encoders can be actively homed with the ‘Home (active)’ command. Configure the command by (cf. Figure 5-24).

1. Select the ‘Home (active)’ command.
2. Select the axis to home.
3. Enter the homing position.
4. Select the command transition.
5. Save the changes.

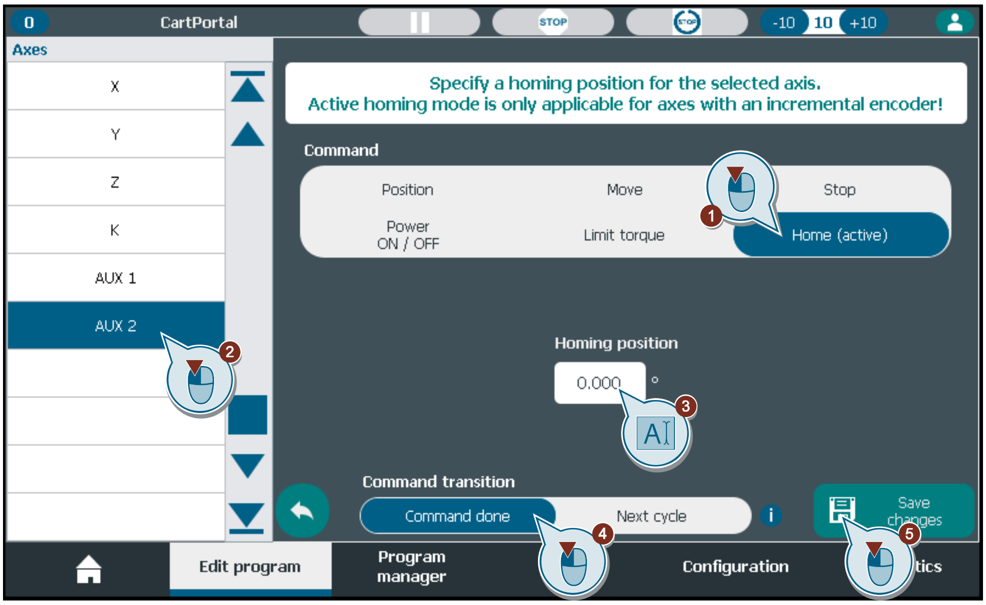
The homing sequence is defined in the TO configuration (cf. chapter 2.3.1).

NOTICE

An active homing command can only be applied to axes with an incremental encoder.

For all other encoders, a configuration error occurs when running the program.

Figure 5-24: Single axis command - Active homing



Specify a homing position for the selected axis.
Active homing mode is only applicable for axes with an incremental encoder!

Homing position: 0.000 °

For more detailed information about single axis functions read the following documentation.

SIMATIC S7-1500 S7-1500/S7-1500T Axis functions V5.0 in TIA Portal V16

Read me

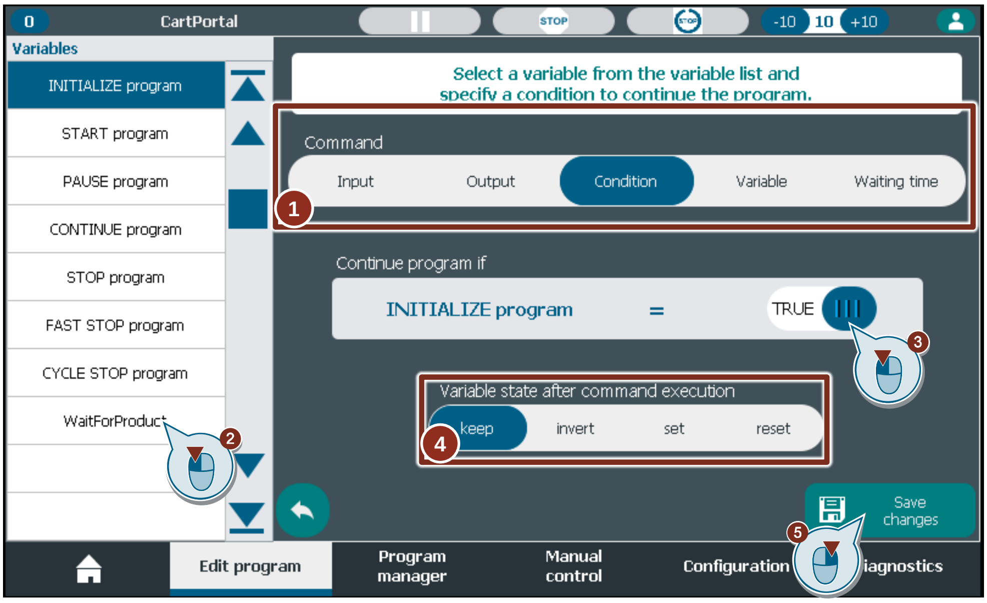
5.5 Wait commands

Wait commands can be used to control the program flow. The program waits with further command execution until the wait condition is fulfilled. Configure the command by (cf. Figure 5-24)

1. Select what to wait for
   - Input
   - Output
   - Condition
   - Variable
   - Waiting time
2. Select an input, an output, a variable, or a condition.
3. Choose the state it must have for the wait command to be fulfilled.
   - For variables "<, <=, =, >, >=" can be chosen for the comparison and a fixed value must be entered.
4. For outputs and conditions: Select the state the output or condition should have after the wait command is fulfilled.
   - 'Keep': Do not change the state.
   - 'Invert': Invert the state.
   - 'Set': Set the state to TRUE.
   - 'Reset': Reset the state to FALSE.
5. Save the changes.

The waiting time can be either defined manually with a constant value or by selecting a floating-point variable that is evaluated at runtime when the command becomes active.

Figure 5-25: Wait command – Wait for condition



Select a variable from the variable list and specify a condition to continue the program.

Command: Condition

Continue program if INITIALIZE program = TRUE

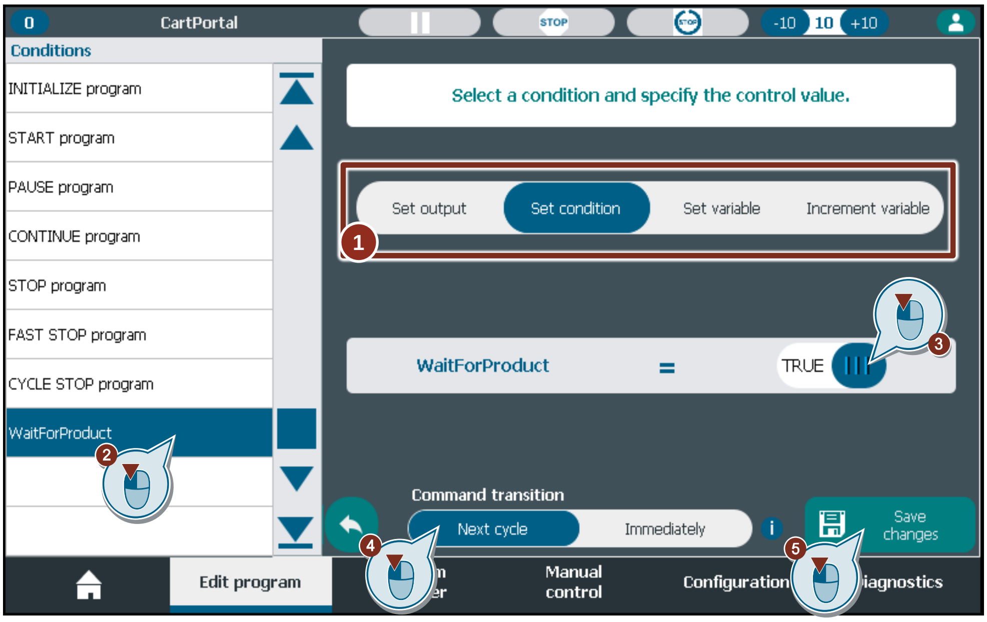
Variable state after command execution: keep

5.6 Set commands

Set commands can be used to change the state of outputs, conditions, or variables. Configure the command by (cf. Figure 5-26)

1. Select what to change
   - Output.
   - Condition.
   - Variable.
2. Select an output, a condition, or a variable.
3. Choose the state or value to change it to or increment it by.
4. Select a command transition.
5. Save the changes.

Figure 5-26: Set command – Set condition



NOTE

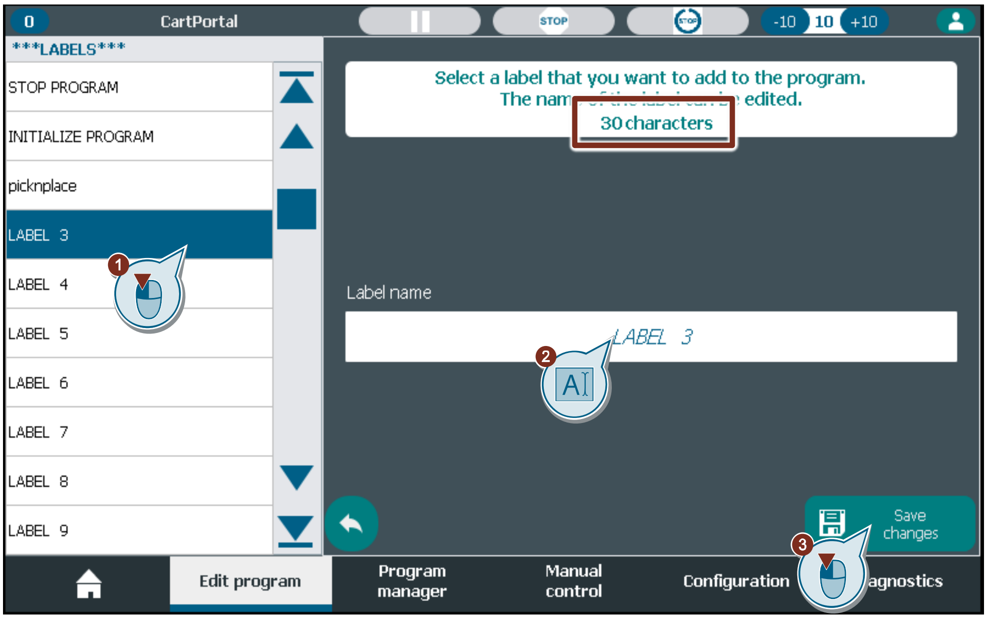
With the command transition ‘Immediately’ multiple commands can be executed in the same PLC cycle. The maximum number of commands per PLC cycle is limited to 10. Executing multiple commands per PLC cycle increases the PLC load.

5.7 Labels

Labels can be used as reference marks for other commands such as jumps, loops, parallel sequences or simply to structure the program visually. A label must always be unique and can therefore only be inserted once in the program. For the same reason a label cannot be copied. Labels can be renamed without having to save the changes for each name adaption. Configure the command by (cf. Figure 5-27)

1. Select a label to insert and/or rename.
2. Enter a label name.
3. Save the changes.

Figure 5-27: Label command



NOTICE	**Deleting a label may affect other commands!** This can change the program flow. If a label is deleted from the program, all commands referring to this label are commented out.

5.8 Calculate command

Variables can be assigned a calculated value using the calculate command. The following formula is available for the calculation.

$$y = a + b*x + c*x^2$$

Configure the command by (cf. Figure 5-28)

1. Select ‘y’.
2. Select a variable for ‘y’.
3. Activate or deactivate a term.
4. Enter a value for b (or a or c).
5. Select ‘x’.
6. Select a variable for ‘x’.
7. Save the changes.

Figure 5-28: Calculate command

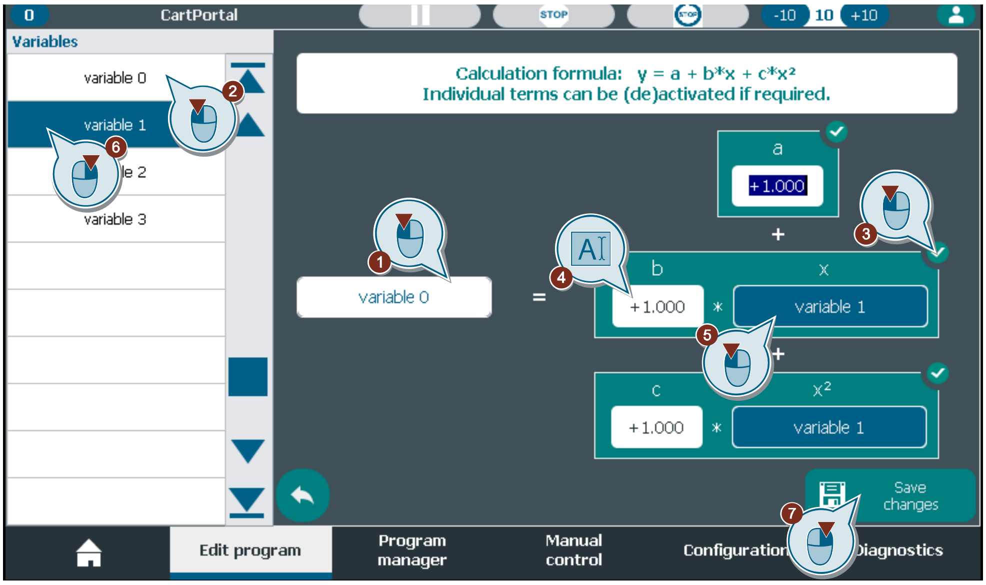


Figure 5-28 results in the following equation:

```
variable 0 = 1 + 1 * variable 1 + 1 * (variable 1)²
```

5.9 Parallel sequence commands

The program execution always starts in the main sequence. If this sequence terminates all sequences are terminated. The SCP offers the possibility to start and stop additional sequences that run in parallel to the main sequence. If these terminate the main sequence and all other active sequences will continue executing. Each active sequence is calculated once per PLC cycle. A parallel sequence is always started at a label with the ‘Start sequence’ command. The sequence is stopped with the ‘Stop sequence’ command or when encountering a ‘Program end’ in the sequence. All parallel sequences are stopped with the ‘Stop all active sequences’ command or when the main sequence encounters the ‘Stop program’ label.

Sequences are independent of program slots and command. That means that multiple sequences can execute the same commands. In general, it is recommended however to separate sequences into their own program slots.

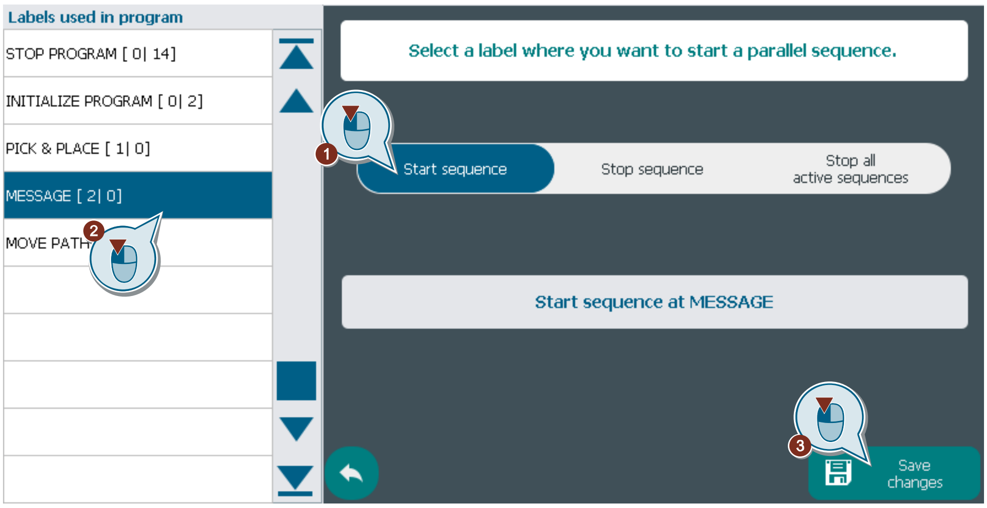
NOTE Multiple active sequences raise the PLC load and thus affect PLC performance.

Start a parallel sequence

A ‘Start sequence’ command is configured by (cf. Figure 5-29)

1. Select the ‘Start sequence’ command.
2. Select the label where to start the sequence (only inserted labels are shown).
3. Save the changes.

Figure 5-29: Insert a parallel sequence command



Labels used in program

STOP PROGRAM [0| 14]

INITIALIZE PROGRAM [0| 2]

PICK & PLACE [1| 0]

MESSAGE [2| 0]

MOVE PATH

Select a label where you want to start a parallel sequence.

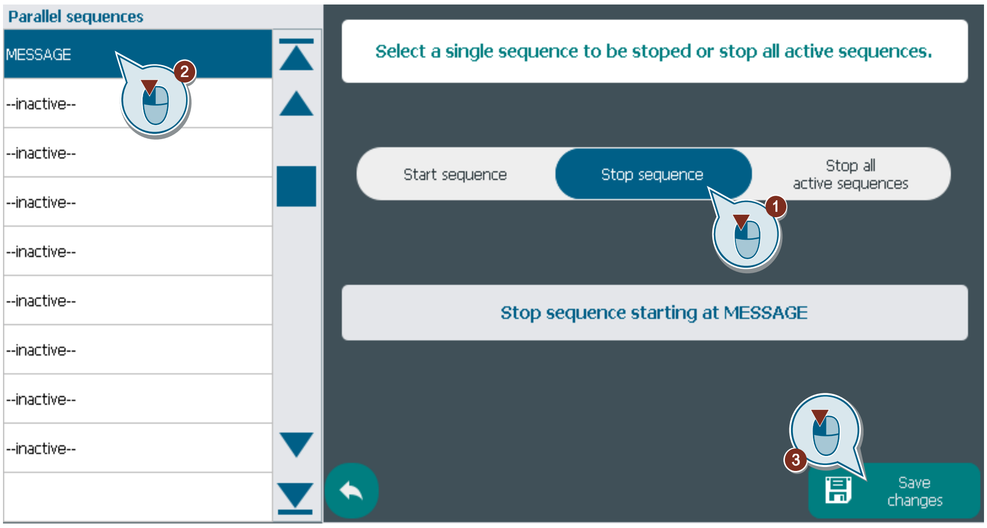
Start sequence at MESSAGE

Stop a parallel sequence or all sequences

A ‘Stop sequence’ command is configured by (cf. Figure 5-30)

1. Select the ‘Stop sequence’ command.
2. Select the sequence to stop. The sequence is identified by the label the sequence started at. Only sequences that are started at some point in the program are shown.
3. Save the changes.

Figure 5-30: Stop a parallel sequence



Parallel sequences

MESSAGE

--inactive--

--inactive--

--inactive--

--inactive--

--inactive--

--inactive--

--inactive--

--inactive--

Select a single sequence to be stoped or stop all active sequences.

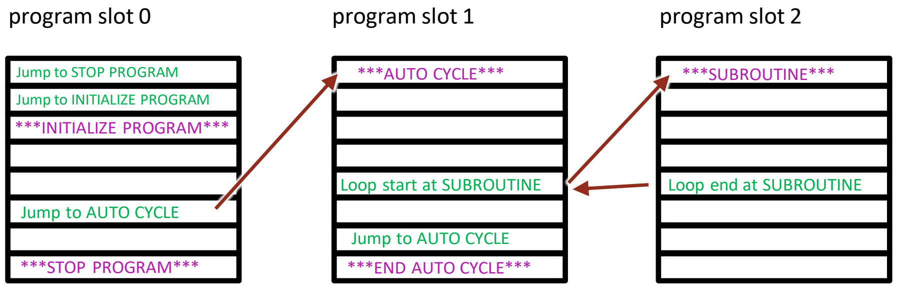
Stop sequence starting at MESSAGE

To stop the entire program, insert the ‘Stop all active sequences’ command. All running sequences including the main sequence will be stopped and the main sequence will execute the stopping routine.

5.10 Jump/Loop commands

Jumps are always combined with a label. Once the jump executes the program execution continues at the referenced label. In the special case of loops the program execution continues after the loop command when all loops are executed. Figure 5-31 illustrates the program flow with a jump and a loop command.

Figure 5-31: Jumps and Loops



Jump

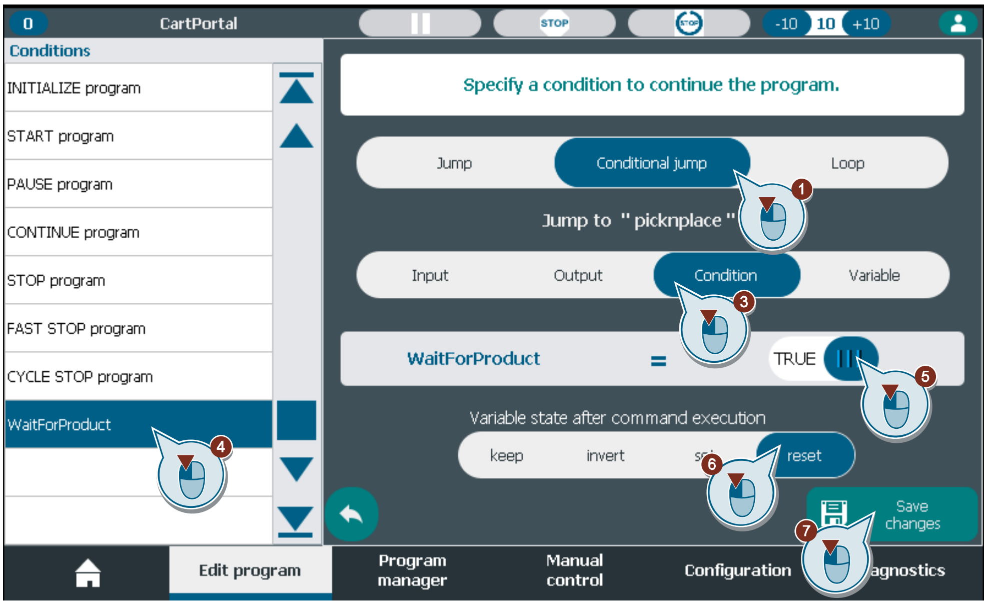
A jump command without a condition will always be executed and continues the program at the referenced label.

Conditional jump

A conditional jump command will be executed if the defined condition is fulfilled. If the condition is not fulfilled, the jump is skipped. Configure the command by (cf. Figure 5-32)

1. Select ‘conditional jump’
2. Select a label to jump to (only inserted labels are shown).
3. Select the condition type (this updates the table on the left)
   - Input.
   - Output.
   - Condition.
   - Variable.
4. Select the input, output, condition, or variable from the table.
5. Select the state or value to check against.
   - For variables “<, <=, =, >, >=” can be chosen for the comparison and a fixed value must be entered.
6. For outputs and conditions: Select the state the output or condition should have after the jump command is executed.
   - ‘Keep’: Do not change the state.
   - ‘Invert’: Invert the state.
   - ‘Set’: Set the state to TRUE.
   - ‘Reset’: Reset the state to FALSE.
7. Save the changes.

Figure 5-32: Conditional jump command

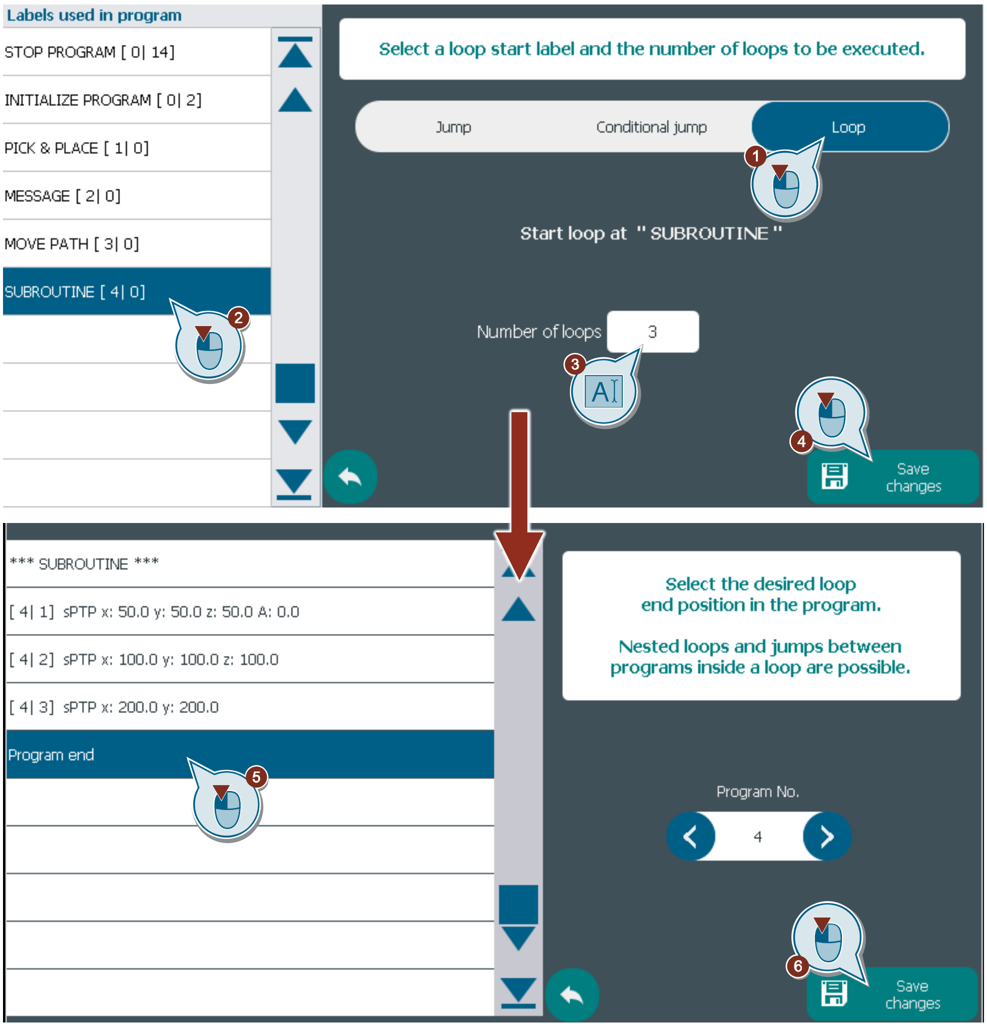


Loop

A loop command first jumps to the configured label. The commands following the label are executed until a Loop End command is found. If several loops are configured, the commands between the start label and the loop end command are executed a given number of times. Afterwards, the loop returns to the loop command and continues with the next command. The loop end will be displayed in a command line. Configure the command by (cf. Figure 5-33)

1. Select the ‘Loop’ command.
2. Select the label to start the loop.
3. Enter the number of loos to execute.
4. Save the changes. Another screen for the loop end will open.
5. Select the end position of the loop.
6. Save the changes again. End loop command will be inserted above the selected line.

Figure 5-33: Loop command



Labels used in program

STOP PROGRAM [0| 14]

INITIALIZE PROGRAM [0| 2]

PICK & PLACE [1| 0]

MESSAGE [2| 0]

MOVE PATH [3| 0]

SUBROUTINE [4| 0]

Select a loop start label and the number of loops to be executed.

Jump | Conditional jump | Loop

Start loop at " SUBROUTINE "

Number of loops: 3

Save changes

*** SUBROUTINE ***

[4| 1] sPTP x: 50.0 y: 50.0 z: 50.0 A: 0.0

[4| 2] sPTP x: 100.0 y: 100.0 z: 100.0

[4| 3] sPTP x: 200.0 y: 200.0

Program end

Select the desired loop end position in the program.

Nested loops and jumps between programs inside a loop are possible.

Program No.: 4

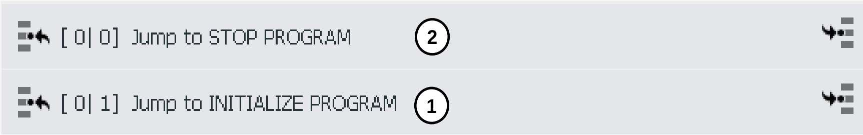
Save changes

6 Programming examples

6.1 Standard program execution

The program execution always starts with the command 'Jump to INITIALIZE PROGRAM' and in case of a stop request of any kind except for the 'Fast stop' (cf. chapter 6.2) the program executes the command 'Jump to STOP PROGRAM'. These commands cannot be changed and therefore the labels 'INITIALIZE PROGRAM' and 'STOP PROGRAM' must be implemented (cf. Figure 6-1).

Figure 6-1: Default commands



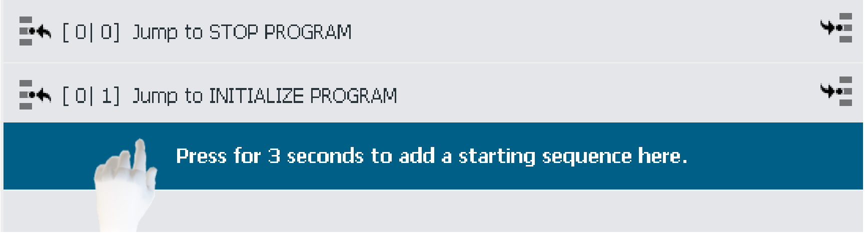
[0| 0] Jump to STOP PROGRAM

[0| 1] Jump to INITIALIZE PROGRAM

6.1.1 Add a starting sequence

To support with the initial program creation a predefined starting sequence can be added in any program slot. This feature is only available if the program is empty. To add the starting sequence press and hold the command for three seconds (cf. Figure 6-2). This adds a sequence of default that include an initialization sequence and a stop sequence.

Figure 6-2: Add default starting sequence



[0| 0] Jump to STOP PROGRAM

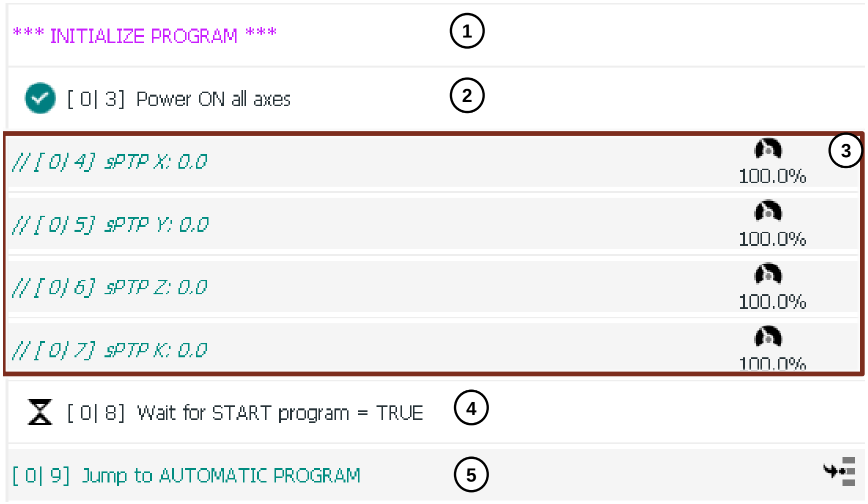
[0| 1] Jump to INITIALIZE PROGRAM

Press for 3 seconds to add a starting sequence here.

6.1.2 Initialize program

The initialization sequence always starts with the label 'INITIALIZE PROGRAM' and ends with the command 'Wait for START program = TRUE'. This means that the initialize button, key or condition must be true until the last command of the initialization sequence is executing. After the last command the automatic program can start. In most cases it is best practice to follow with a jump command to a label in a separate program slot and create the automatic program there. Figure 6-3 shows the initialization sequence that is created with the default starting sequence (cf. chapter 6.1.1) which consists of the 'INITIALIZE PROGRAM' label (1), a power on of all axes (2) a commented out initial positioning sequence of all kinematics axes (3) and the 'Wait for START program = TRUE' command (4). Lastly, a jump command to the automatic program sequence was added to illustrate the best practice (5). The positioning sequence of the kinematics axes adapts to the selected kinematics type. In general, this sequence should be adapted to the use case at hand.

Figure 6-3: Default initialization sequence



```
*** INITIALIZE PROGRAM ***
✓ [ 0| 3]  Power ON all axes
// [ 0| 4]  sPTP X: 0.0                100.0%
// [ 0| 5]  sPTP Y: 0.0                100.0%
// [ 0| 6]  sPTP Z: 0.0                100.0%
// [ 0| 7]  sPTP K: 0.0                100.0%
⧗ [ 0| 8]  Wait for START program = TRUE
[ 0| 9]  Jump to AUTOMATIC PROGRAM
```

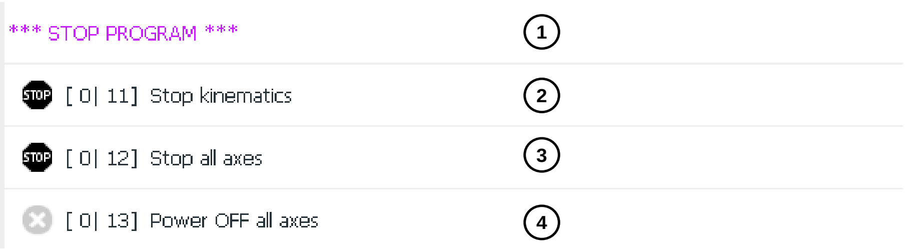
6.1.3 Stop program

Various events trigger a stop of the automatic program.

- The first ‘Program end’ command that is processed in the main sequence.
- Stop all sequences command is processed.
- Jump command to ‘STOP PROGRAM’ label
- Stop button is pressed at the home screen or at the head bar.
- The condition ‘STOP program’ is set.
- An error occurred during program execution.

All these triggers lead to the execution of the user defined stopping sequence. A trigger leads to a jump to the ‘STOP PROGRAM’ label and executes the subsequent commands. Figure 6-4 shows the stopping sequence that is created with the default starting sequence (cf. chapter 6.1.1) which consists of the ‘STOP PROGRAM’ label (1), a ‘Stop kinematics’ command that stops with the current dynamics (2), a ‘Stop all axes’ command that stops with the default dynamics (3) and a power off of all axes (3). The program execution ends with the command ‘Program end’. The user can expand this sequence as desired and add other necessary commands, such as resetting outputs or moving axes to a specific position. It is recommended to do this after the axes were stopped and disabled.

Figure 6-4: Default stopping sequence



```
*** STOP PROGRAM ***
STOP [ 0| 11]  Stop kinematics
STOP [ 0| 12]  Stop all axes
✕ [ 0| 13]  Power OFF all axes
```

⚠ CAUTION

Multiple errors in the program execution

The first error always triggers the execution of the stopping sequence even if the error occurs while executing the stopping sequence. Subsequent errors are added to the diagnostics buffer (cf. chapter 8.2) and the execution continues with the next command and is terminated with a ‘Program end’ command.

6.2 Fast stop program

The condition 'FAST STOP program' can be used to stop the automatic program with maximum dynamics in an emergency, such as triggering an emergency stop button or a light curtain. Independent of the user defined stop sequence the fast stop sequence consists of the commands

1. Stop kinematics with emergency deceleration
2. Stop all axes with emergency deceleration
3. Power off all axes.

The fast stop sequence is not visible in the program table and cannot be changed. It can be executed at every point of the automatic program if it is running even in the stop sequence.

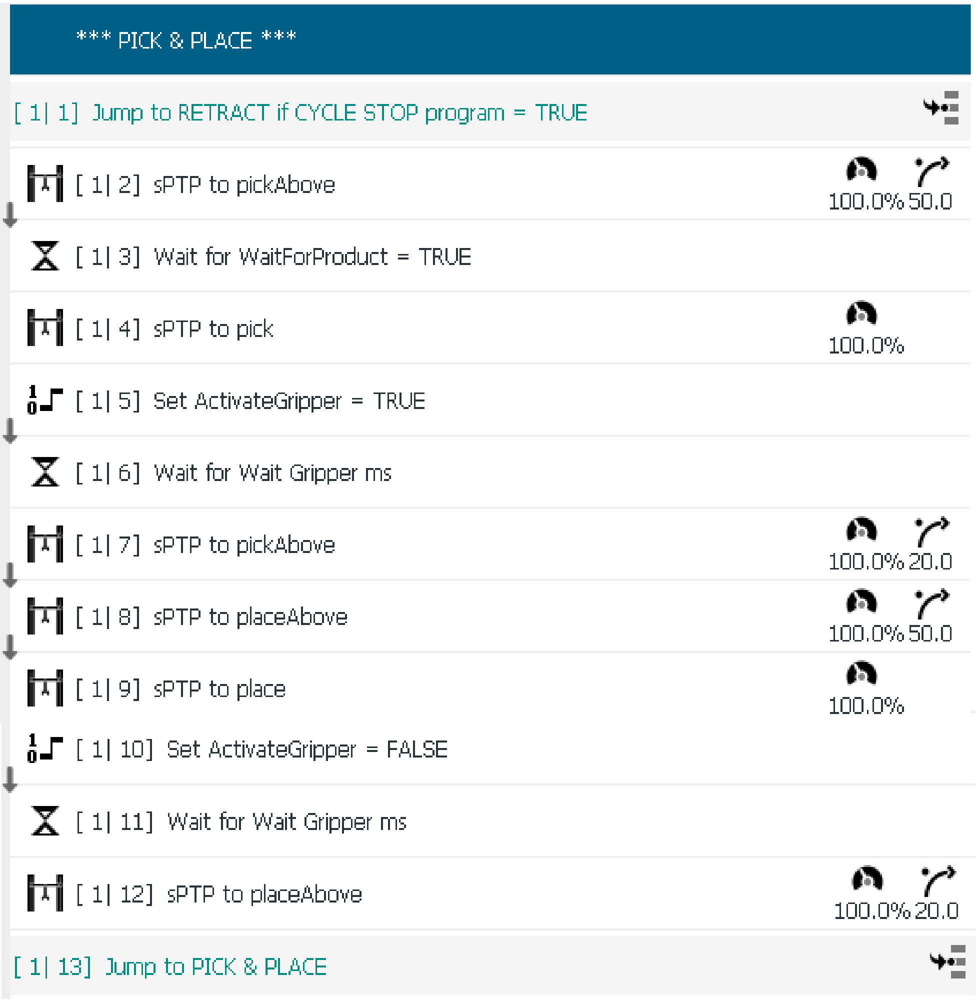
NOTE

The condition 'FAST STOP program' can be provided with a logic in the corresponding configuration mask (cf. chapter 3.7.4) to connect all emergency stop triggers to this condition.

6.3 Pick & Place program

The command sequence for a simple Pick & Place program is shown in Figure 6-5. The pick and place subprogram starts with a label as a reference to call the subprogram. The kinematics motion commands are referenced to points from the point table (cf. chapter 4.4). The kinematics waits at the ‘pickAbove’ point if no product is available. At the ‘pick’ position the gripper is activated and the program execution waits for a short time defined by the variable ‘Wait Gripper’. At the place position the gripper is deactivated and the program pauses again defined by ‘Wait Gripper’. With the jump command the sequence is executed indefinitely. If possible, the kinematics motion is blended to achieve a better cycle time. The example project also implements a cycle stop that is described in chapter 6.4.

Figure 6-5: Example program for a pick and place program



*** PICK & PLACE ***

[1| 1] Jump to RETRACT if CYCLE STOP program = TRUE

[1| 2] sPTP to pickAbove 100.0% 50.0

[1| 3] Wait for WaitForProduct = TRUE

[1| 4] sPTP to pick 100.0%

[1| 5] Set ActivateGripper = TRUE

[1| 6] Wait for Wait Gripper ms

[1| 7] sPTP to pickAbove 100.0% 20.0

[1| 8] sPTP to placeAbove 100.0% 50.0

[1| 9] sPTP to place 100.0%

[1| 10] Set ActivateGripper = FALSE

[1| 11] Wait for Wait Gripper ms

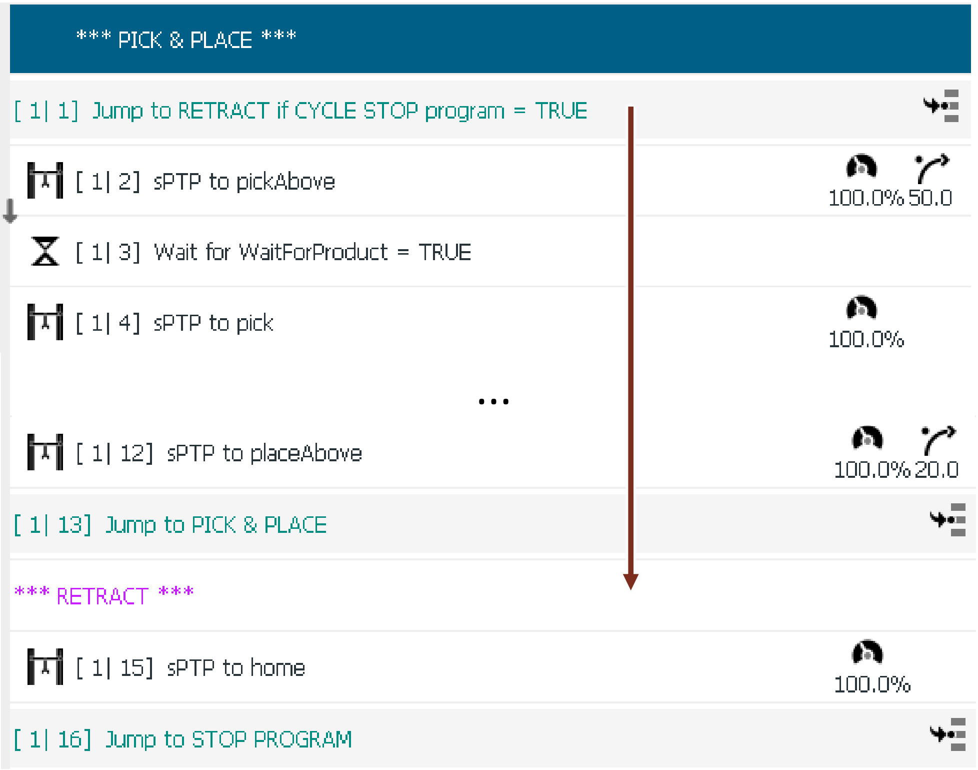
[1| 12] sPTP to placeAbove 100.0% 20.0

[1| 13] Jump to PICK & PLACE

6.4 Cycle stop program

The condition ‘CYCLE STOP program’ can be used to stop the automatic program execution in a controlled manner at a defined position in the program. The condition needs to be used in the program in conjunction with a conditional jump that stops or suspends the program, usually at the beginning or end of a machine cycle. The conditional jump can be present in multiple parallel sequences. An example for the use of the cycle stop is shown in Figure 6-6. At the beginning of a new Pick & Place cycle, there is a conditional jump for the condition ‘CYCLE STOP program’ that jumps to the program label ‘RETRACT’ when the condition is TRUE. After the ‘RETRACT’ label the kinematics will be moved to a ‘Home’ position and the program execution jumps to the default ‘STOP PROGRAM’ label, which terminates the program execution. In this way it is possible to stop the machine in a controlled manner during production or testing.

Figure 6-6: Example program for cycle stop in a pick and place sequence

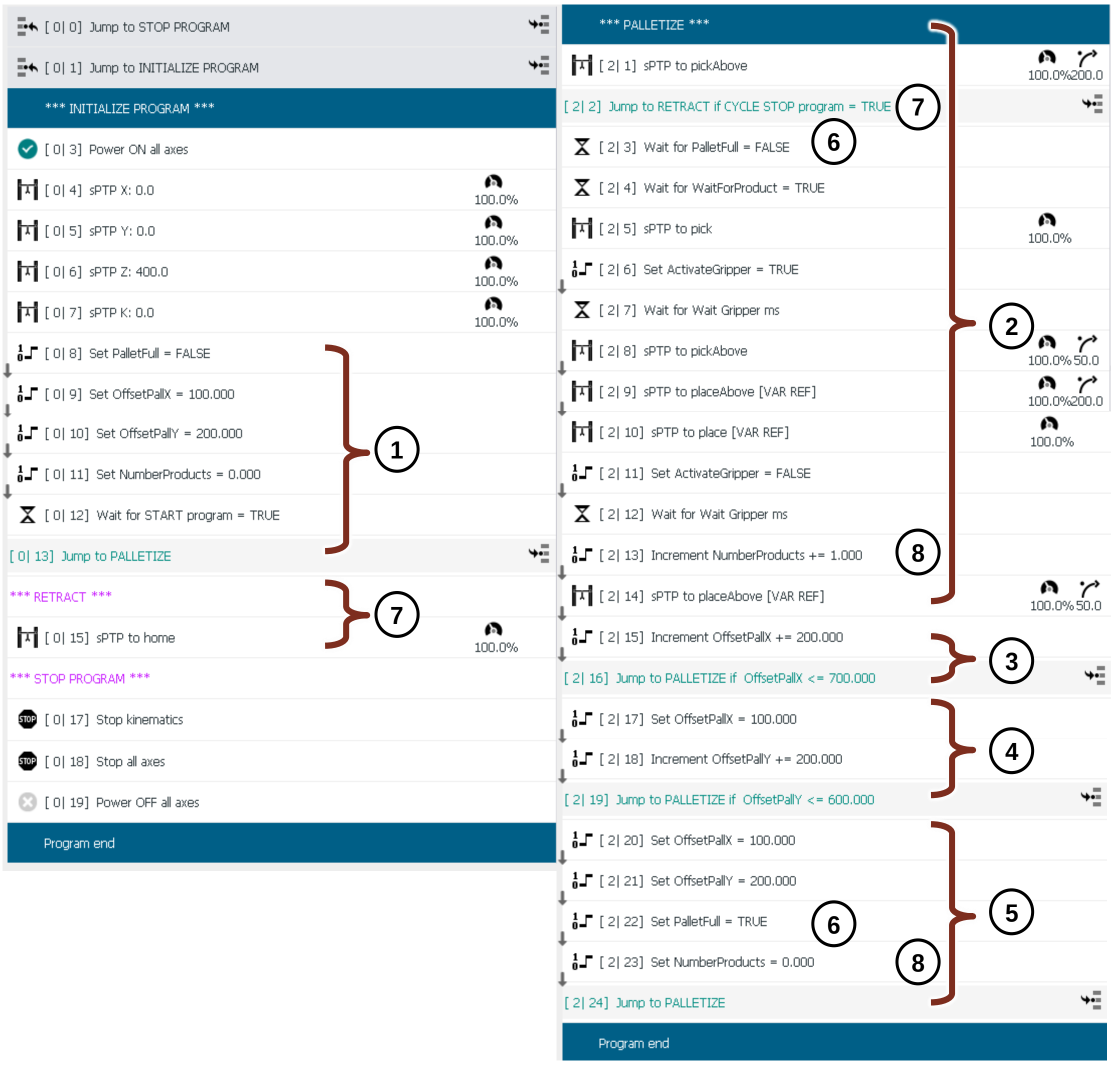


```
*** PICK & PLACE ***
[ 1| 1]  Jump to RETRACT if CYCLE STOP program = TRUE
[ 1| 2]  sPTP to pickAbove                      100.0% 50.0
[ 1| 3]  Wait for WaitForProduct = TRUE
[ 1| 4]  sPTP to pick                           100.0%
                                  ...
[ 1| 12]  sPTP to placeAbove                    100.0% 20.0
[ 1| 13]  Jump to PICK & PLACE
*** RETRACT ***
[ 1| 15]  sPTP to home                          100.0%
[ 1| 16]  Jump to STOP PROGRAM
```

6.5 Palletizing program

A palletizing program can be created in many ways depending on the pattern at hand. Figure 6-7 shows an example for a rectangular regular pattern where the pick position does not change. On the left side is program slot 0. The default starting sequence (cf. chapter 6.1.1) was added and extended to fit the need of the application. The initialization of the necessary variables and conditions (1) as well as the jump to the palletizing sequence in program slot 1 on the right were added. The sequence starts with a pick and place sequence (2) like in chapter 6.3. The difference is that the 'placeAbove' and 'place' positions were combined with the variables 'OffsetPalX' and 'OffsetPalY' such that these positions can change during runtime and less position commands are needed. The first row of the palletizing pattern extends in the x-direction and is achieved by incrementing the 'OffsetPalX' by 200mm (2). If the 'OffsetPalX' is less than 700mm, which translates to four products per row the palletizing continues. After four products the 'OffsetPalX' is reset and the 'OffsetPalY' is incremented by 200mm (4) such that palletizing of the second column starts. If the 'OffsetPalY' is less than 600mm, which translates to three columns, the palletizing continues. This results in twelve products on the pallet. After this both offsets are reset such that the palletizing can start over again (5). The demonstrate an operator interaction the condition 'PalletFull' is also set and the palletizing only continues if this variable is manually reset (6). To achieve this the condition is connected to a message as a trigger variable (cf. Figure 6-8). When the variable is set the message appears on the home screen and the operator can change the pallets. Afterwards the operator confirms that a new pallet is ready, and the palletizing sequence continues. As seen in chapter 6.4 a cycle stop was also included (7) such that a controlled stop of the machine process is possible. Lastly a product counter was added for each pallet for information purposes. This palletizing subroutine can easily be extended in the z-direction to stack several layers on the pallet with an additional variable 'OffsetPalZ'. Figure 6-9 shows the resulting path of the palletizing program.

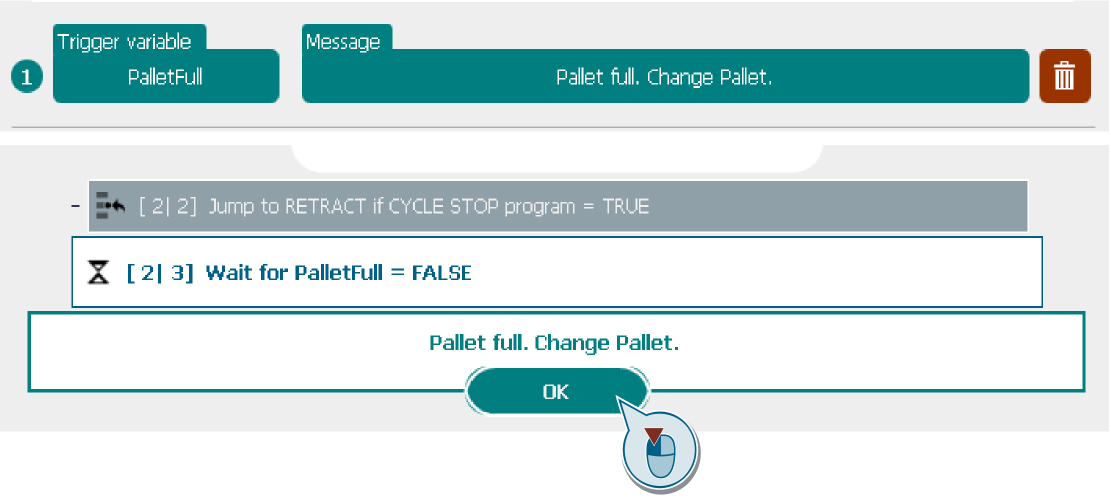
Figure 6-7: Palletizing example - Program with point and variable references in program slot 0 and 1



[0| 0] Jump to STOP PROGRAM
[0| 1] Jump to INITIALIZE PROGRAM
*** INITIALIZE PROGRAM ***
[0| 3] Power ON all axes
[0| 4] sPTP X: 0.0 — 100.0%
[0| 5] sPTP Y: 0.0 — 100.0%
[0| 6] sPTP Z: 400.0 — 100.0%
[0| 7] sPTP K: 0.0 — 100.0%
[0| 8] Set PalletFull = FALSE
[0| 9] Set OffsetPallX = 100.000
[0| 10] Set OffsetPallY = 200.000
[0| 11] Set NumberProducts = 0.000
[0| 12] Wait for START program = TRUE
[0| 13] Jump to PALLETIZE
*** RETRACT ***
[0| 15] sPTP to home — 100.0%
*** STOP PROGRAM ***
[0| 17] Stop kinematics
[0| 18] Stop all axes
[0| 19] Power OFF all axes
Program end

*** PALLETIZE ***
[2| 1] sPTP to pickAbove — 100.0% 200.0
[2| 2] Jump to RETRACT if CYCLE STOP program = TRUE
[2| 3] Wait for PalletFull = FALSE
[2| 4] Wait for WaitForProduct = TRUE
[2| 5] sPTP to pick — 100.0%
[2| 6] Set ActivateGripper = TRUE
[2| 7] Wait for Wait Gripper ms
[2| 8] sPTP to pickAbove — 100.0% 50.0
[2| 9] sPTP to placeAbove [VAR REF] — 100.0% 200.0
[2| 10] sPTP to place [VAR REF] — 100.0%
[2| 11] Set ActivateGripper = FALSE
[2| 12] Wait for Wait Gripper ms
[2| 13] Increment NumberProducts += 1.000
[2| 14] sPTP to placeAbove [VAR REF] — 100.0% 50.0
[2| 15] Increment OffsetPallX += 200.000
[2| 16] Jump to PALLETIZE if OffsetPallX <= 700.000
[2| 17] Set OffsetPallX = 100.000
[2| 18] Increment OffsetPallY += 200.000
[2| 19] Jump to PALLETIZE if OffsetPallY <= 600.000
[2| 20] Set OffsetPallX = 100.000
[2| 21] Set OffsetPallY = 200.000
[2| 22] Set PalletFull = TRUE
[2| 23] Set NumberProducts = 0.000
[2| 24] Jump to PALLETIZE
Program end

Figure 6-8: Palletizing example - Message configuration and display on the home screen

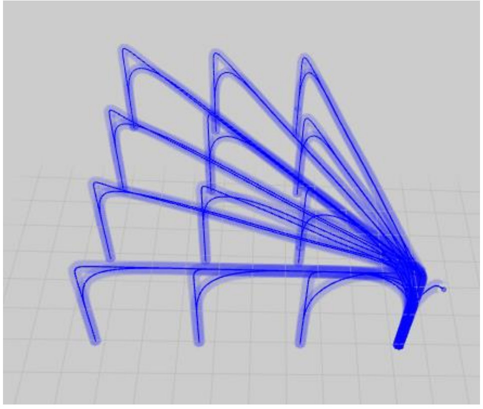


1 Trigger variable: PalletFull — Message: Pallet full. Change Pallet.

[2| 2] Jump to RETRACT if CYCLE STOP program = TRUE
[2| 3] Wait for PalletFull = FALSE

Pallet full. Change Pallet.

Figure 6-9: Palletizing example – Resulting path



7 Operate the machine

7.1 Machine control

The automatic program execution can be controlled via

- The home screen.
- The head bar.
- The function keys.
- The default conditions.

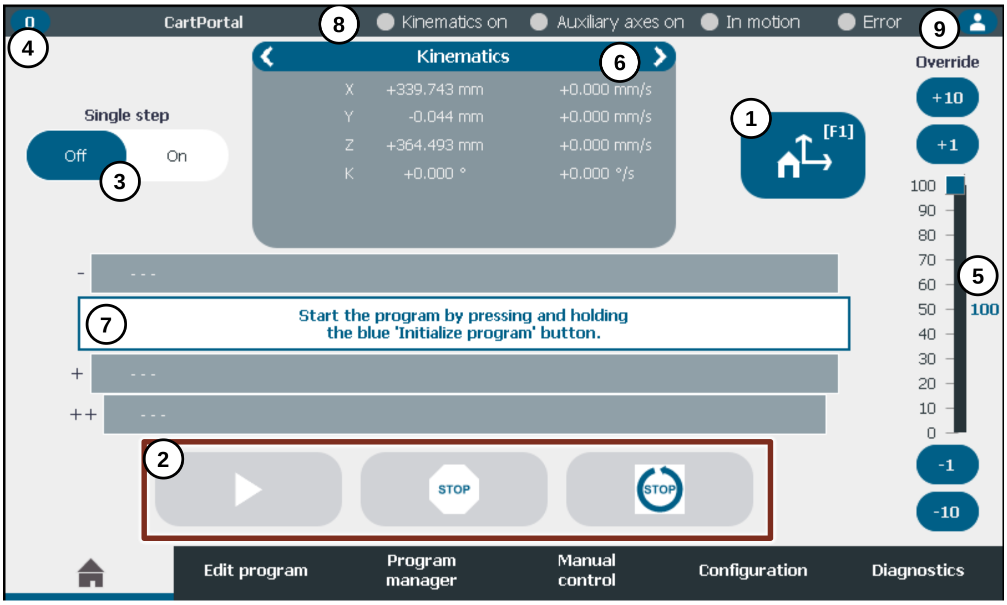
The automatic program can only be started in the operation modes “Full operation” and “Automatic” (cf. chapter 2.11).

7.1.1 Machine control via home screen

The home screen offers the user the possibility to operate the machine with a user interface. The operating functions available are listed below (cf. Figure 7-1):

1. Start program with initialization sequence.
2. Start / Stop / Cycle stop / Pause / Continue machine program.
3. Switch the execution mode to single step.
4. Channel selection.
5. Change of the velocity override.
6. Display of kinematics and axes positions and user-defined buttons.
7. Display currently executed command, previous and two following commands.
8. Program name and status.
9. User login.

Figure 7-1: Overview of the SIMATIC Kinematics Integrator home screen



X	+339.743 mm	+0.000 mm/s
Y	-0.044 mm	+0.000 mm/s
Z	+364.493 mm	+0.000 mm/s
K	+0.000 °	+0.000 °/s

Start the program by pressing and holding the blue 'Initialize program' button.

Machine start

If the single step is turned off the machine program must be started in two stages. First, the initialize button (1) must be pressed and held to start the initialization program. When the button is released before the initialization finished, a stop will be initiated. After the initialization is executed, the play button (2 - left) becomes active (blue). The automatic program can now be started by clicking this button. If the single step is turned on the initialization sequence can be stepped through with the single step button that is visible instead of the play button. The initialize button is inactive (grey). For detailed information about the command sequence of the initialization program, refer to chapter 6.1.2.

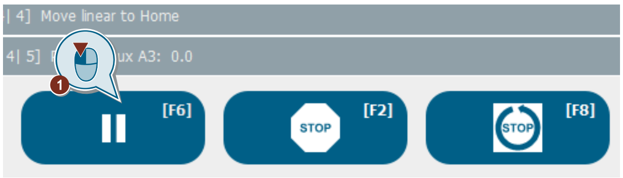
Machine stop

When the program is running, the stop button (2 - middle) becomes active. By pressing the stop button, the active program is terminated by running through the stopping routine (cf. chapter 6.1.3).

Pause program execution

When the program is running, the pause button (cf. Figure 7-2) becomes visible and active. By pressing the pause button, the active program sequences are interrupted at the after the currently executing command.

Figure 7-2: Pause program execution

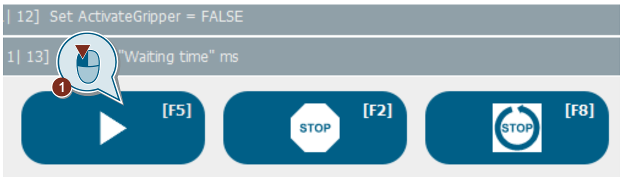


⚠ **CAUTION**	**PAUSE program does not interrupt motion commands** If the program is paused motion commands will not be interrupted but finished first. A ‘Move velocity’ command will not be interrupted at all. All axes will continue their motion until their target positions are reached. A running sPTP command will execute without taking the configured blending into the next sPTP command into account. In case all axis motion shall be interrupted, the override on the home screen can be set to 0. This does not stop the program execution of non-motion commands.

Continue program execution

A paused program can be continued if the play button is pressed again.

Figure 7-3: Pause program execution



Cycle stop program

If the cycle stop is configured in the program, it becomes active after the initialization sequence is executed (cf. chapter 6.4). Once the button has been pressed, the background color changes to green, which indicates the condition ‘CYCLE STOP program’ has been set and the cycle stop is pending (cf. Figure 7-4 and Figure 7-5). After the cycle stop is executed, the condition resets automatically. The cycle stop can be pressed again to reset the ‘CYCLE STOP program’ condition. The button turns blue again. If no cycle stop is configured in the program the button is disabled (grey).

Figure 7-4: Cycle stop program inactive

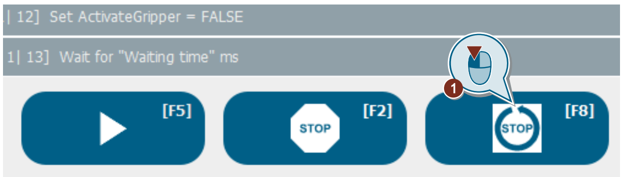
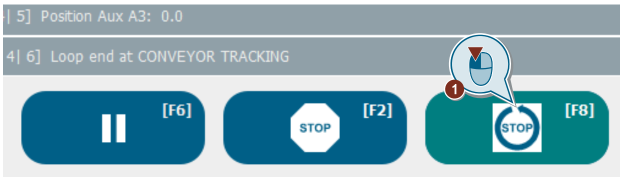

| 12] Set ActivateGripper = FALSE
1| 13] Wait for "Waiting time" ms
[F5]
STOP
[F2]
1
STOP
[F8]

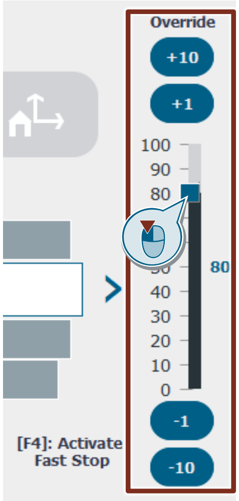
Figure 7-5: Cycle stop program pending


| 5] Position Aux A3: 0.0
4| 6] Loop end at CONVEYOR TRACKING
[F6]
STOP
[F2]
1
STOP
[F8]

Change override

With the override slider on the right of the screen, the speed of the kinematics and axes can be changed as a percentage value (cf. Figure 7-6). While the automatic program is running the override can only be changed with the button '+/- 1/10'. The override only affects the velocity.

Figure 7-6: Override slider and buttons



Single step

When single step is turned on, the program can be executed in a sequential order (cf. Figure 7-7). The play button will be replaced by the single step button. When another step of the program can be executed, the button becomes active. Each click of the single step button executes the next command in every active sequence. The 'Home' tab only shows the main sequence. All active sequences can be seen in the 'Edit program' tab (cf. chapter 7.3).

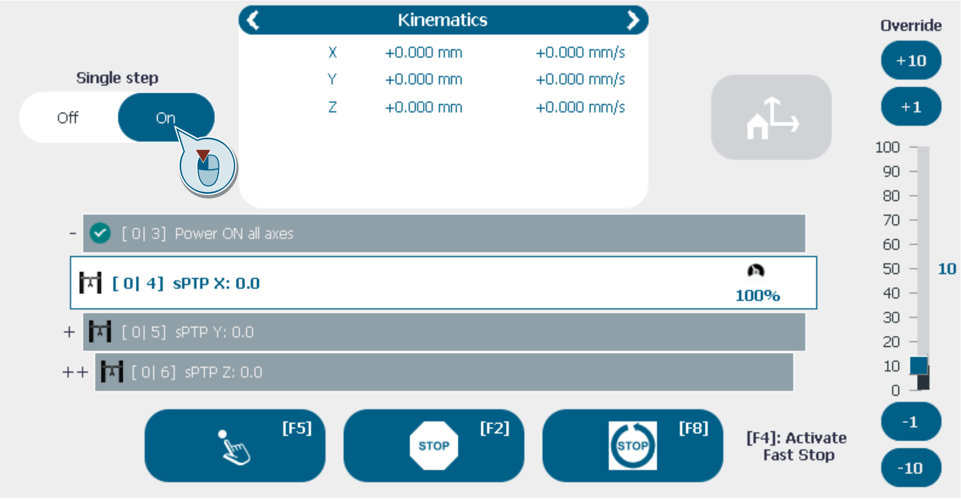
CAUTION

Blending with sPTP

If an sPTP command is configured with blending the next single step can be executed when the blending segment is reached. Without another sPTP step the kinematics moves to the target position. This can lead to different paths in single step mode compared to automatic mode.

Use blocked zones and make sure the target positions are never in collision. When in doubt use additional points in the blending segment. Additionally, always test the program in automatic mode with a low velocity override.

Figure 7-7: Single step switch



Error handling

Errors are displayed on the home screen with basic information. Check the ‘Diagnostics’ tab of the SIMATIC Cartesian Portal to get more detailed information on the error (cf. chapter 8.2).

7.1.2 Machine control via head bar

The head bar on the HMI has the following functions in different screens:

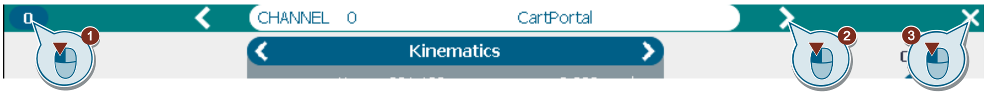
- Channel selection (all screens).
- Display of diagnostic data (home screen).
- User login (all screens).
- Program control and override setting if home screen is not selected.

Channel selection

The channel selection can be entered from every screen if at least two channels are configured. The select a channel (cf. Figure 7-8):

1. Enable the channel selection by clicking on the channel number in the top left corner.
2. Cycle through the configured channels with the arrow buttons.
3. Close the channel selection with the ‘X’ in the top right corner or by clicking on the channel number in the top left corner again.

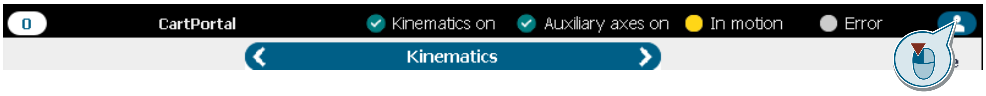
Figure 7-8: Head bar - Channel selection



User login

The user login in the head bar can be accessed in every screen by clicking the button in the top right corner (cf. Figure 7-9). The channel selection must be deactivated.

Figure 7-9: Head bar - User login



Diagnostics data

The diagnostics data in the head bar is visible if the channel selection is deactivated and the home screen is currently active. The following states are displayed (cf.Table 7-1).

Table 7-1: Head bar – Diagnostics data

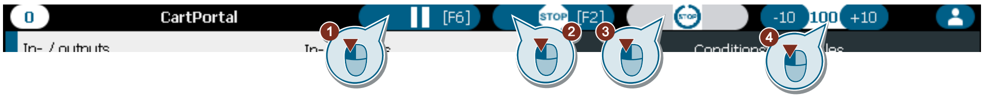
State	Description
Kinematics on	All kinematic axes are enabled.
Auxiliary axes on	All additional axes are enabled.
In motion	At least one axis is moving.
Error	Error in program or technology object.

Program control and override

The program control and override are active if the home screen is currently inactive, and the channel selection is deactivated. The buttons in the head bar have the same functionality as the control elements on the home screen (cf. Figure 7-10). However only pausing or stopping elements for the program execution are present in the head bar.

1. Pause the program.
2. Stop the program.
3. Cycle stop the program.
4. Change the velocity override.

Figure 7-10: Program control in head bar



NOTE The cycle stop is only active if the initialization sequence is finished, and the cycle stop is configured (cf. chapter 6.4).

NOTE The override cannot be changed in the manual control tab since the override control is taken by this tab.

7.1.3 Machine control via function keys

If the panel supports functions keys and they are activated the program can also be controlled via the these (cf. chapter 2.12). By default, the functions keys that lead to the execution of commands are only active in the home screen whereas function keys that stop the execution of commands are active in every screen. Their functionality is shown in Table 7-3.

NOTE If the function keys are enabled the buttons on the home screen and in the head bar show the corresponding functions keys.

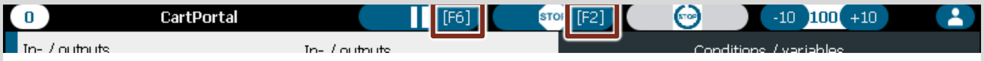


Table 7-2: Function keys for machine control

Function Key	Description	Active
F1	Initialize the program	Home screen
F2	Stop the program	Always
F3	Start the program	Home screen
F4	Fast stop the program	Always
F5	Continue/Single step the program	Home screen
F6	Pause the program	Always
F8	Activate the cycle stop	Always

7.1.4 Machine control externally via default conditions

It is possible to control the SIMATIC Cartesian Portal program with the default conditions in the 'LSCP_GlobalProjectData'. Use the conditions of the 'LSCP_GlobalProjectData.glProjectData[channelIndex]' in Table 7-3 to control the program execution without the HMI.

Table 7-3: Conditions for external program control

Variable	Description
conditions[0]	Hold TRUE: Program start stage 1: INITIALIZE program
conditions[1]	Set TRUE: Program start stage 2: START program
conditions[2]	Set TRUE: PAUSE program execution (interrupts kinematics motions)
conditions[3]	Set TRUE: CONTINUE program execution
conditions[4]	Set TRUE: STOP program
conditions[5]	Toggle TRUE: FAST STOP program (with emergency deceleration)
conditions[6]	Toggle TRUE: CYCLE STOP program (cycle stop must be programmed)

7.2 Edit program while program execution is paused

During pause or single-step mode the user can execute the following actions:

- Configure or adapt point positions in the point table
- Jog axes that are not enabled in the program with the manual control
- Edit, add, or delete commands with the program editor

⚠ CAUTION

Modifying, adding, or deleting commands or points during pause can lead to unexpected behavior of the machine.

The program sequences will be continued at their current command number. If commands are added or deleted the program can continue with the unexpected commands.

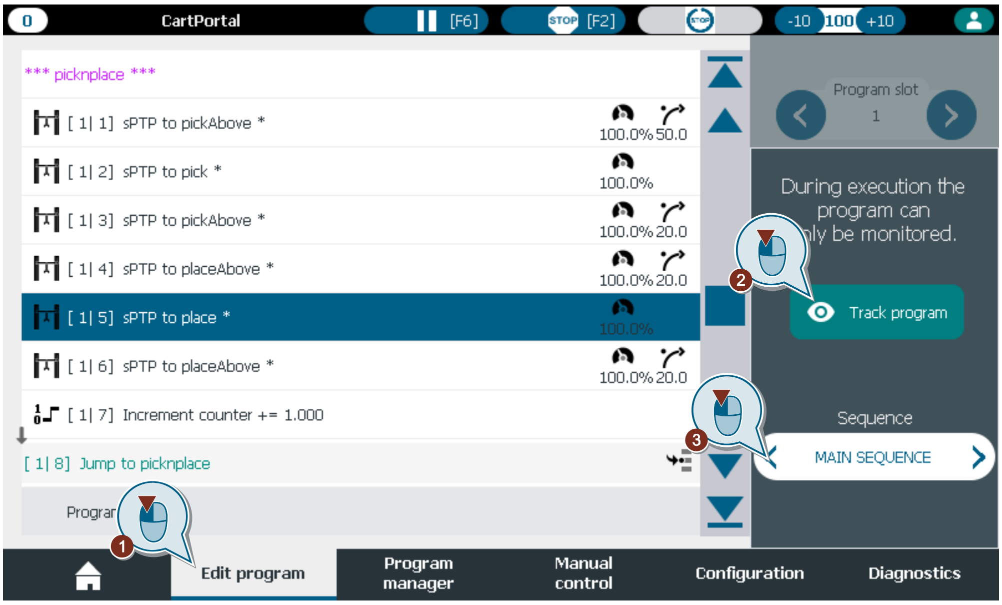
It is not recommended to add or delete commands in pause mode. Only minor changes of existing commands or points should be executed during pause or single-step mode.

7.3 Track program

Besides the home screen that shows the main sequence the program editor enables tracking of all active sequence when an automatic program is running. To track the program (cf. Figure 7-11):

1. Select the ‘Edit program’ tab.
2. Click the ‘Track program’ button.
3. Cycle through the active sequences with the arrow buttons.

Figure 7-11: Track program in ‘Edit program’



*** picknplace ***

[1| 1] sPTP to pickAbove * 100.0% 50.0

[1| 2] sPTP to pick * 100.0%

[1| 3] sPTP to pickAbove * 100.0% 20.0

[1| 4] sPTP to placeAbove * 100.0% 20.0

[1| 5] sPTP to place * 100.0%

[1| 6] sPTP to placeAbove * 100.0% 20.0

[1| 7] Increment counter += 1.000

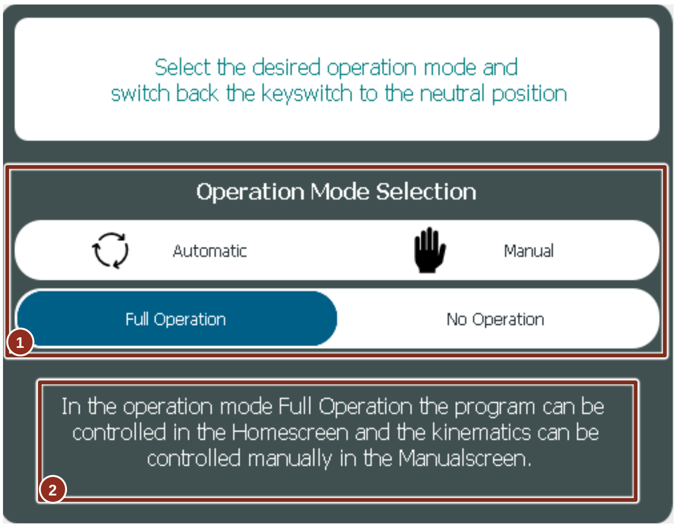
[1| 8] Jump to picknplace

During execution the program can only be monitored.

7.4 Operation modes

Operation modes can be changed on the HMI. When using a mobile panel with a key switch, the operation mode selection is already built-in and can be accessed by turning the key switch into a position. This opens a pop up where the operation modes can be selected. At startup of the PLC the operation mode is set to default to ‘Full Operation’.

Figure 7-12: Operation mode pop up



Select the desired operation mode and switch back the keyswitch to the neutral position

Operation Mode Selection

In the operation mode Full Operation the program can be controlled in the Homescreen and the kinematics can be controlled manually in the Manualscreen.

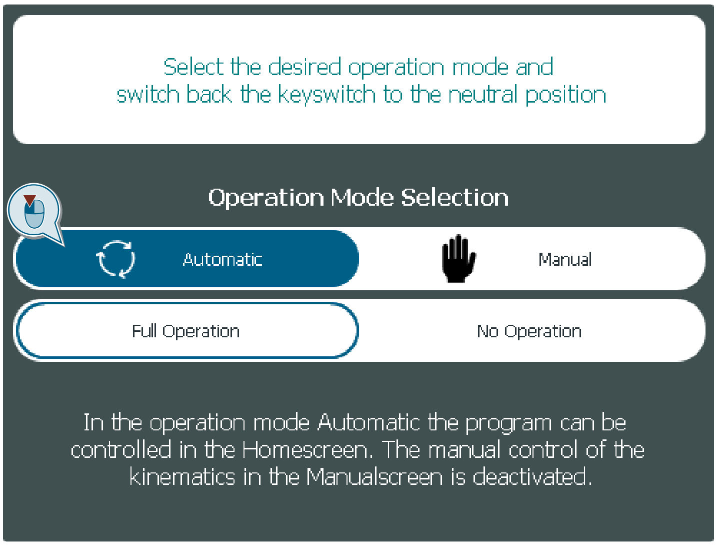
The operation mode pop up contains a header with a general description for the selection of operation modes, the operation mode selection (2) and a description for each operation mode (3) (cf. Figure 7-12).

Operation mode selection

The operation mode can be changed by turning the key switch (or setting the respective HMI interface variable, see chapter 2.11). The operation mode pop up will appear and a desired operation mode can be selected.

Click on the desired operation mode to preselect it. The preselected operation mode is indicated by a blue background. The currently active operation mode is indicated by the blue frame. The activate the preselected operation mode turn the key switch back or press the respective HMI interface variable again. In Figure 7-13 the ‘Full operation’ operation mode is currently active while the ‘Automatic’ operation mode is preselected. Upon turning the key switch again, the ‘Automatic’ operation mode will become active.

Figure 7-13: Operation mode selection

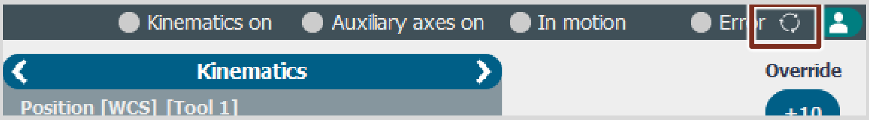


Select the desired operation mode and switch back the keyswitch to the neutral position

In the operation mode Automatic the program can be controlled in the Homescreen. The manual control of the kinematics in the Manualscreen is deactivated.

NOTE

The active operation mode for the operation modes ‘Automatic’ and ‘Manual’ are displayed with an icon in the head bar. The icon is identical to the icon in the pop up.

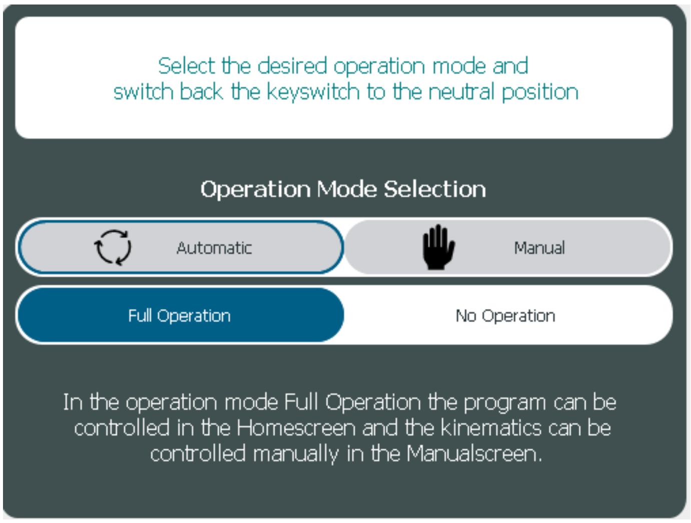


User level for operation modes

The selection of an operation mode can be coupled to a user login level. The configuration of the user levels for each operation mode is described in chapter 2.11. The user can only select an operation mode that requires a user level of the currently logged in user or lower.

Figure 7-14 shows that the currently active operation mode is ‘Automatic’ (blue frame). The ‘Automatic’ and ‘Manual’ operation mode cannot be selected due to an insufficient user level (grey background). The user preselected the ‘Full Operation’ operation mode (blue background) and could also preselect the ‘No Operation’ operation mode. If the key switch is turned again the ‘Full Operation’ operation mode would become active and the currently logged in user would not be able to switch back to ‘Automatic’. Note that the currently active operation mode can always be preselected again independent of the user level such that no change in operation mode occurs.

Figure 7-14: Operation modes with user level requirement



Select the desired operation mode and switch back the keyswitch to the neutral position

Operation Mode Selection

In the operation mode Full Operation the program can be controlled in the Homescreen and the kinematics can be controlled manually in the Manualscreen.

8 Diagnostics

The diagnostics screen shows the state of various system components

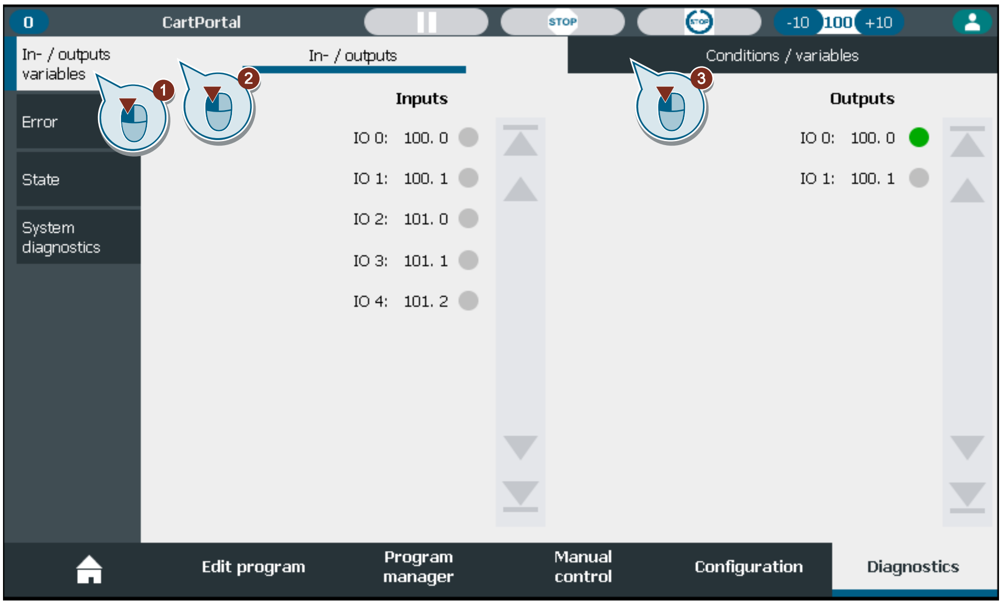
- Actual state of inputs and outputs.
- Actual state of conditions and variables.
- State of the program and pending errors.
- Status and error word of the axes.
- Status of the zones.
- System diagnosis.

8.1 In-/ Outputs and variables

The ‘In- / outputs variables’ tab offers information about all configured inputs, outputs, conditions, and variables of the current channel. To navigate (cf. Figure 8-1)

1. Select the ‘In- / outputs variables’ tab.
2. Select the ‘‘In- / outputs’ tab or
3. Select the ‘Conditions / Variables’ tab.

Figure 8-1: In-/ outputs variables



Actual state of inputs and outputs

The status of all configured inputs and outputs is displayed. A green indicator shows that the input or output is TRUE and a grey indicator signals the FALSE (cf. Figure 8-1).

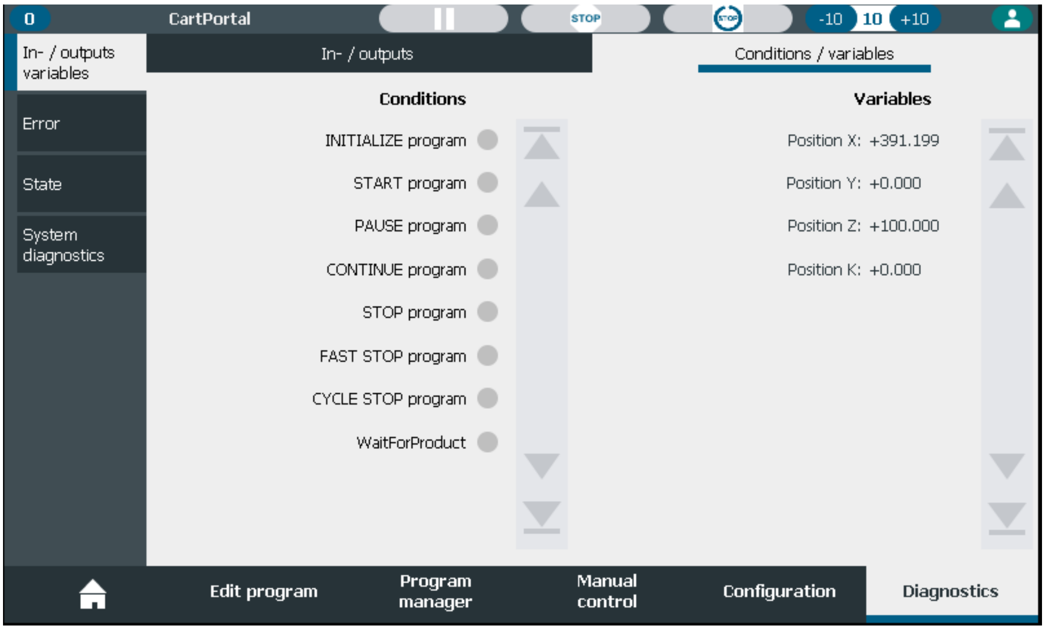
Actual state of conditions and variables

The status of all configured conditions and variables is displayed. A green indicator shows that the condition is TRUE and a grey indicator signals the FALSE (cf. Figure 8-2).

Some conditions and variables are predefined by the SCP (cf. chapter 3.7).

NOTE Inputs, outputs, conditions, and variables are only shown up to the last configured variable.

Figure 8-2: In-/ outputs variables - conditions and variables



Conditions

INITIALIZE program ●
START program ●
PAUSE program ●
CONTINUE program ●
STOP program ●
FAST STOP program ●
CYCLE STOP program ●
WaitForProduct ●

Variables

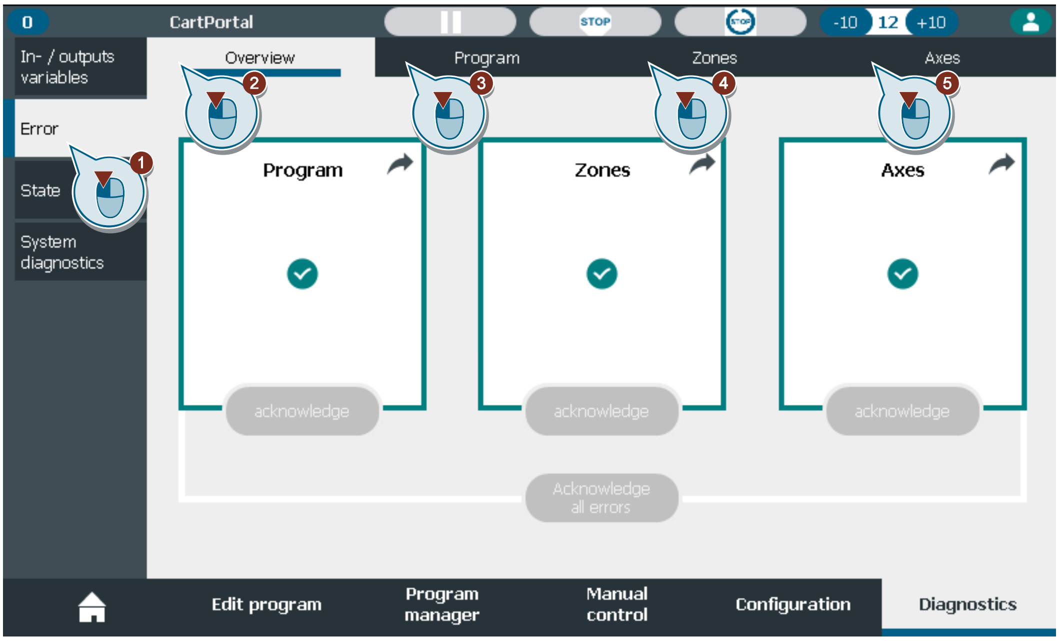
Position X: +391.199
Position Y: +0.000
Position Z: +100.000
Position K: +0.000

8.2 Error

The ‘Error’ tab offers error information about all components of the SCP. To navigate (cf. Figure 8-3)

4. Select the ‘Error’ tab.
5. Select the ‘Overview’ tab or
6. Select the ‘Program’ tab or
7. Select the ‘Zones’ tab or
8. Select the ‘Axes’ tab.

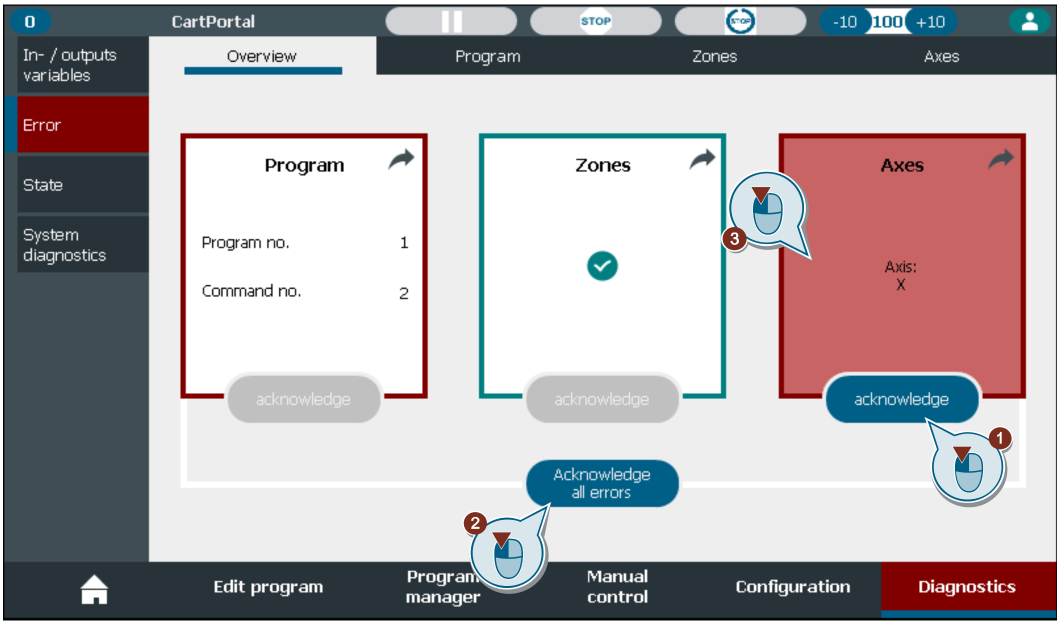
Figure 8-3: Error overview



Overview

The overview shows all three components of the SCP that can throw an error and their state. An error of a component is indicated by a red frame around that component. In the frame additional information like the axis or zone name that has a pending error or the program and command no. where the error occurred during execution. An error of a component can be acknowledged (1) or all errors can be acknowledged at once (2). The acknowledgement of all components follows the order ‘axes’ – ‘zones’ – ‘program’. If errors are acknowledged one at a time this order must be followed. The component that must be acknowledged first is indicated by a red background. To get further information on an error click on the corresponding component (3) (cf. Figure 8-4).

Figure 8-4: Error – Overview



Program no. 1

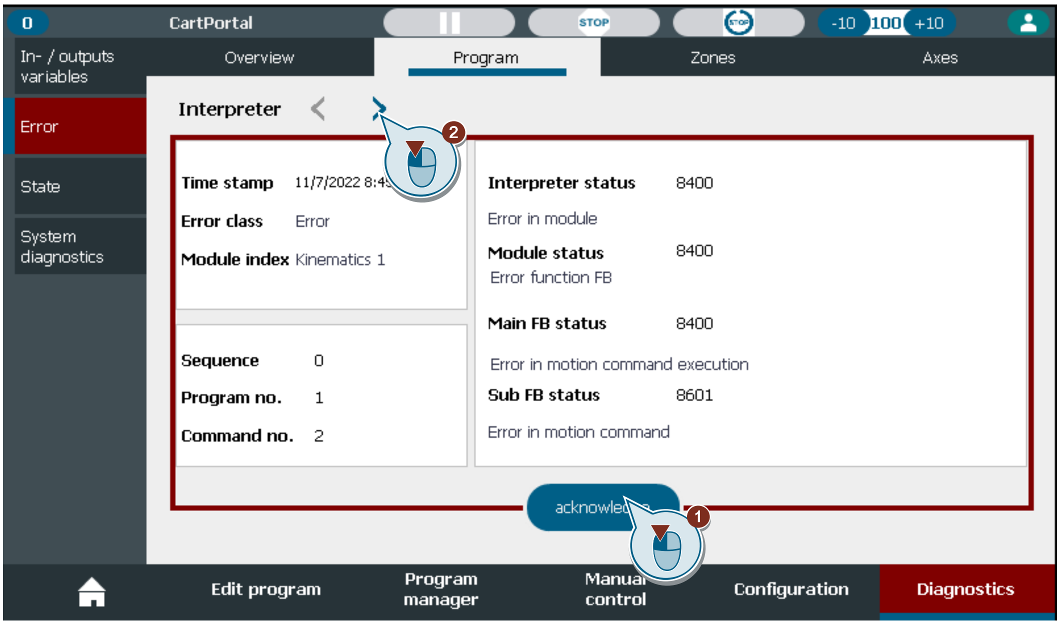
Command no. 2

Axis: X

Program

In the ‘Program’ tab errors that occurred during program execution are displayed. The time stamp, error class and module are displayed. Additionally, the sequence, program number and command number of the command that threw the error is shown. The program error can be acknowledged here if no axis or zone error is pending (1). If multiple errors occurred during program execution, the buttons can be used to navigate through these (2) (cf. Figure 8-5).

Figure 8-5: Error - Program



Time stamp 11/7/2022 8:4

Error class Error

Module index Kinematics 1

Sequence 0

Program no. 1

Command no. 2

Interpreter status 8400

Error in module

Module status 8400

Error function FB

Main FB status 8400

Error in motion command execution

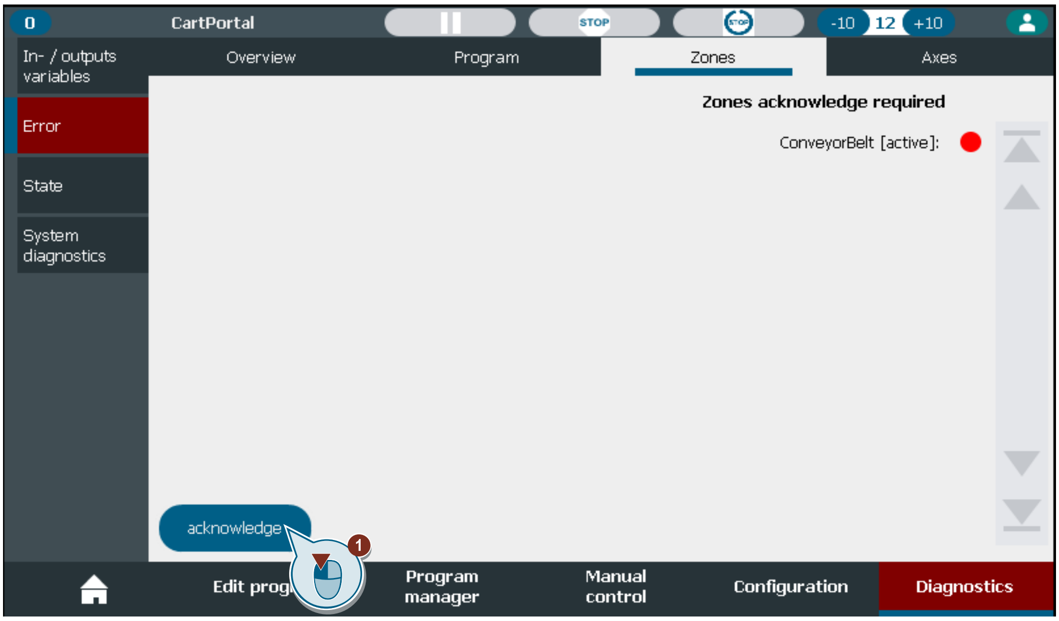
Sub FB status 8601

Error in motion command

Zone error

The ‘Zones’ tab shows all zones that require an acknowledgment. They can be acknowledged directly (1). If an acknowledgement is required, the indicator is red. If no acknowledge is required or the zone is inactive the indicator is grey (cf. Figure 8-6).

Figure 8-6: Error- zones



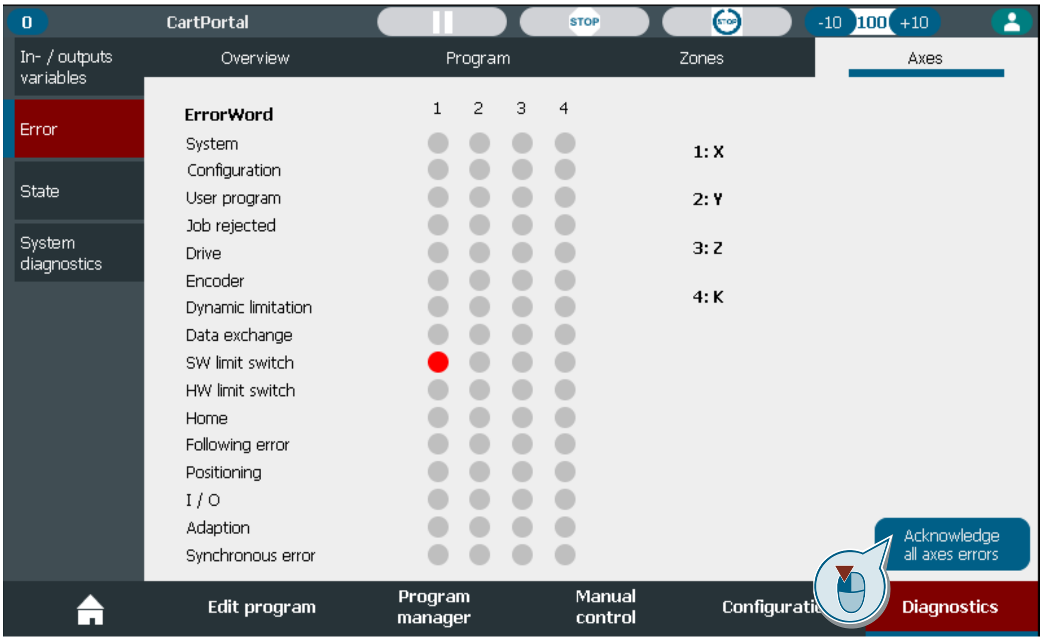
Zones acknowledge required

ConveyorBelt [active]: ●

Axes

The ‘Axes’ tab shows a list of possible TO error that can be pending at each axis. An error is shown by a red indicator. All axes’ errors can be acknowledged at once (cf. Figure 8-7).

Figure 8-7: Axes error word



ErrorWord	1	2	3	4
System	○	○	○	○
Configuration	○	○	○	○
User program	○	○	○	○
Job rejected	○	○	○	○
Drive	○	○	○	○
Encoder	○	○	○	○
Dynamic limitation	○	○	○	○
Data exchange	○	○	○	○
SW limit switch	●	○	○	○
HW limit switch	○	○	○	○
Home	○	○	○	○
Following error	○	○	○	○
Positioning	○	○	○	○
I / O	○	○	○	○
Adaption	○	○	○	○
Synchronous error	○	○	○	○

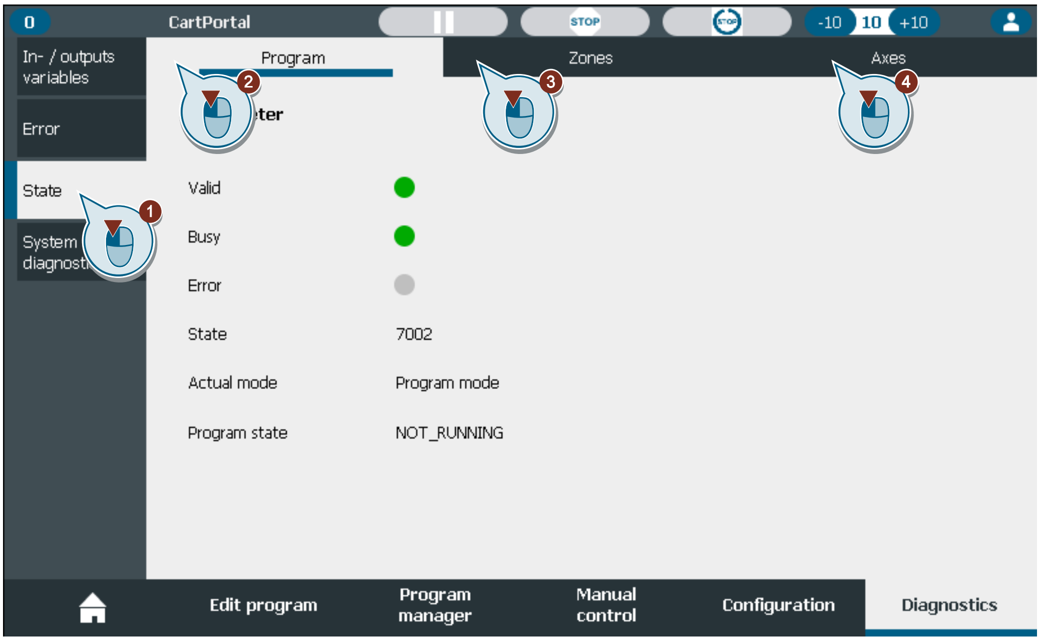
1: X
2: Y
3: Z
4: K

8.3 State

The ‘State’ tab offers state information about all components of the SCP. To navigate (cf. Figure 8-8)

1. Select the ‘State’ tab.
2. Select the ‘Program’ tab or
3. Select the ‘Zones’ tab or
4. Select the ‘Axes’ tab.

Figure 8-8: State overview

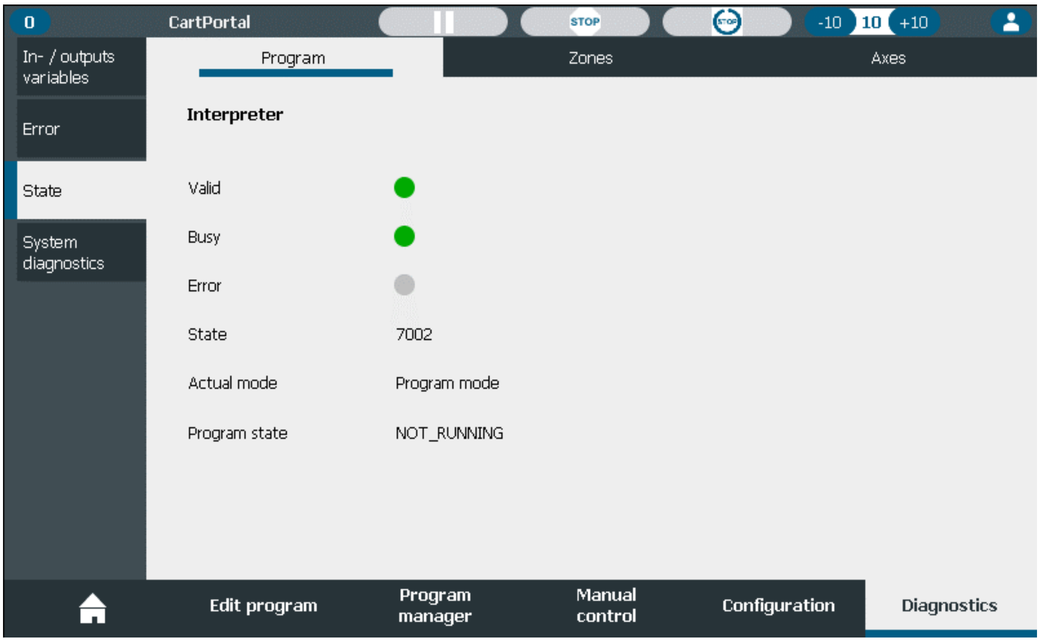


Valid	●
Busy	●
Error	○
State	7002
Actual mode	Program mode
Program state	NOT_RUNNING

Program

The status of the internal interpreter is displayed (cf. Figure 8-9).

Figure 8-9: State - program



Interpreter

Valid ●

Busy ●

Error ●

State 7002

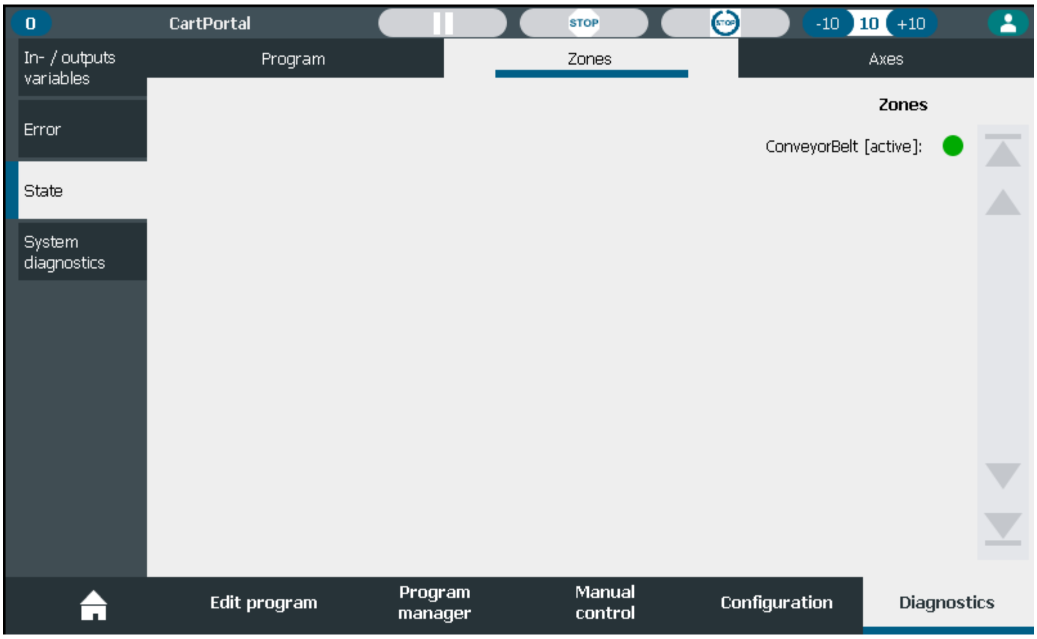
Actual mode Program mode

Program state NOT_RUNNING

Zones

The ‘Zones’ tab shows all zones until the last active one. The zone state is displayed with a red indicator for ‘currently in collision’, a green indicator for ‘currently not in collision’ and a grey indicator for ‘inactive (cf. Figure 8-10).

Figure 8-10: State - zones



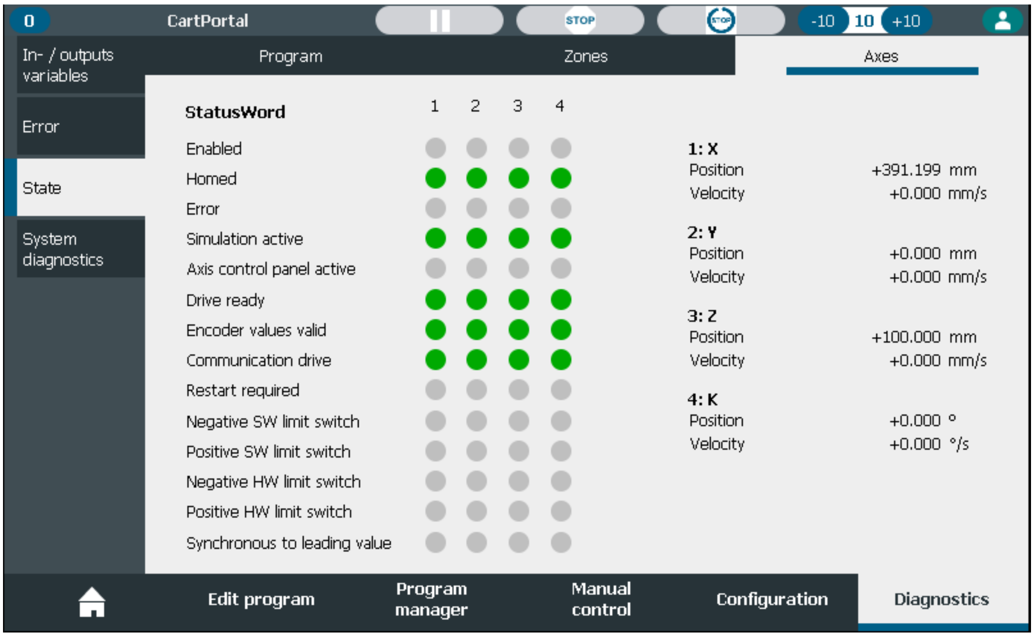
Zones

ConveyorBelt [active]: ●

Axes

The ‘Axes’ tab shows a list of possible TO states of each axis. If the state is active, it is shown by a green indicator otherwise the indicator is grey. Additionally, the current position and velocity of each axis is shown (cf. Figure 8-11).

Figure 8-11: State - axes



StatusWord	1	2	3	4
Enabled	○	○	○	○
Homed	●	●	●	●
Error	○	○	○	○
Simulation active	●	●	●	●
Axis control panel active	○	○	○	○
Drive ready	●	●	●	●
Encoder values valid	●	●	●	●
Communication drive	●	●	●	●
Restart required	○	○	○	○
Negative SW limit switch	○	○	○	○
Positive SW limit switch	○	○	○	○
Negative HW limit switch	○	○	○	○
Positive HW limit switch	○	○	○	○
Synchronous to leading value	○	○	○	○

1: X
Position +391.199 mm
Velocity +0.000 mm/s

2: Y
Position +0.000 mm
Velocity +0.000 mm/s

3: Z
Position +100.000 mm
Velocity +0.000 mm/s

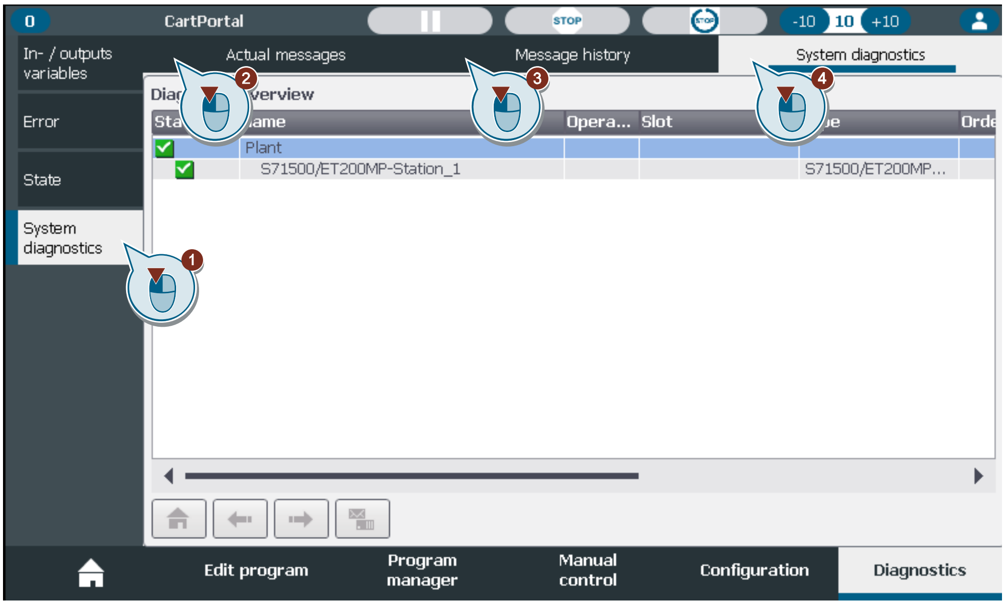
4: K
Position +0.000 °
Velocity +0.000 °/s

8.4 System diagnostics

The ‘System diagnostics’ tab offers system information. To navigate (cf. Figure 8-12)

5. Select the ’System diagnostics’ tab.
6. Select the ‘Actual messages’ tab or
7. Select the ‘Message history’ tab or
8. Select the ‘System diagnostics’ tab.

Figure 8-12: System diagnostics



Diag... overview

Sta	...ame	Opera...	Slot	...e	Orde
✓	Plant				
✓	S71500/ET200MP-Station_1			S71500/ET200MP...	

9 Save and load programs and configurations

9.1 Overview

Concept

You can use the program manager to save and load the following SIMATIC Kinematics Integrator data into or from a csv-file.

- Programs with all commands and labels from the program editor
- Point table
- Input, output, and complete variable configuration
- Current Boolean variable states and variable states
- Home positions and jog velocities

The faceplate works with visual basic scripts and saves these data as csv-files. A csv-file contains the data for one channel.

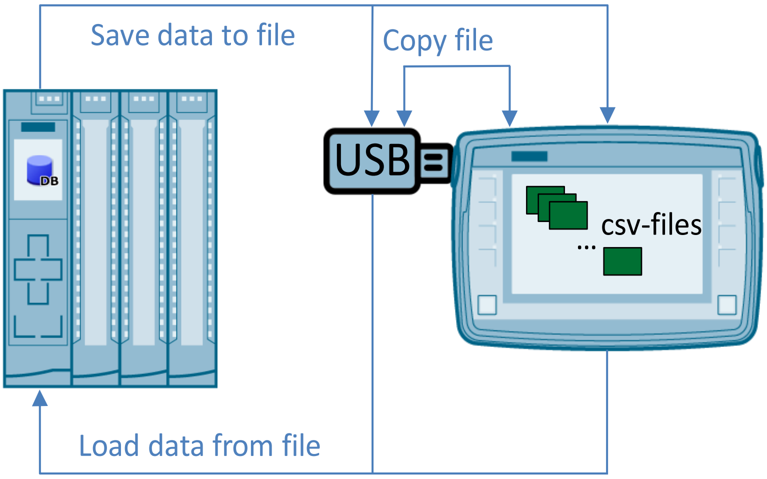
The current data in the PLC are stored in the DB ‘LSCP_GlobalProjectData[channelNumber]’. You can save these data directly to the internal storage of the HMI or to an USB flash drive (cf. Figure 9-1). The number of storable csv-files is limited by the disc capacity of the HMI or a USB flash drive. The csv-file size depends on the project scaling (cf. chapter 2.5) and the number of programmed commands.

Table 9-1 shows a rough estimation for the maximum storable csv-file number. The number can vary depending on the size of the DB ‘LSCP_GlobalProjectData’.

Table 9-1: Number of storable csv-files

Storage	Estimated number of storable csv-files
Internal Storage Mobile Panel 9’	400
USB flash (8GB)	5000

Figure 9-1: Program manager overview



Corrupt program files will be saved if the char ‘;’ is used in Strings.

CAUTION

Loading corrupt csv-files leads to an inconsistent PLC program.

Please do not use the char ‘;’ for point names, messages, variables, and all other names which can be assigned on the HMI.

Memory locations

The memory locations of the csv-files are predefined and not changeable (cf. Table 9-2). The folder ‘ProgramTables’ will be created automatically after the first save or copy program command.

Table 9-2: Memory locations

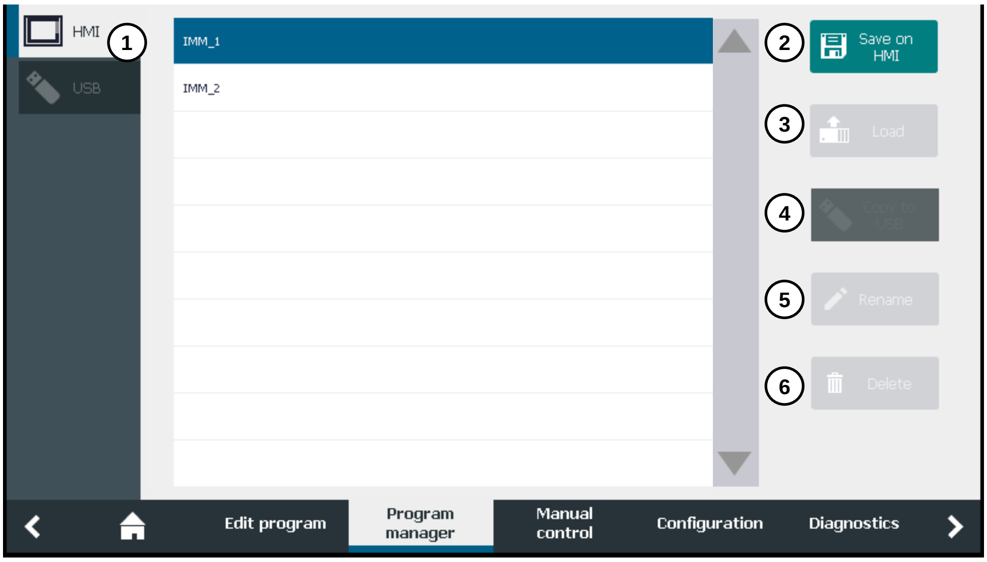
Device	Memory location
PLC	DB “LSCP_GlobalProjectData”
Mobile panel (internal storage)	/flash/ProgramTables/
Comfort panel (internal storage)	/flash/ProgramTables/
USB	/ProgramTables/

9.2 Operation

The program manager screen consists of (cf. Figure 9-2):

1. Navigation: Switch between internal HMI storage or USB (if plugged).
2. Save to HMI/USB: Save current program from PLC to HMI or USB as csv-file. A free spot will be searched automatically. You do not have to mark a free program spot before you press the button ‘Save on HMI’.
   The program will be saved with the current program name.
3. Load: Load marked program to PLC (only possible when the automatic cycle is inactive). The whole actual program with all data will be overwritten.
4. Copy to USB/HMI: Copy the marked program between USB and HMI.
5. Rename: Rename the program name of the marked program.
6. Delete: The marked program will be deleted.

Figure 9-2: Program manager screen



NOTICE

The whole HMI is locked during the load and save process. You cannot operate other channels or faceplates on the HMI.

You cannot stop or start programs of other channels during the loading or saving process.

Make sure, that you do not have to operate the HMI during the safe and load process.

NOTE

The programs for different channels are displayed in the same folder. Make sure that you load the correct programs to the corresponding channel. Define unambiguous program names to avoid loading programs to the wrong channel.

WARNING

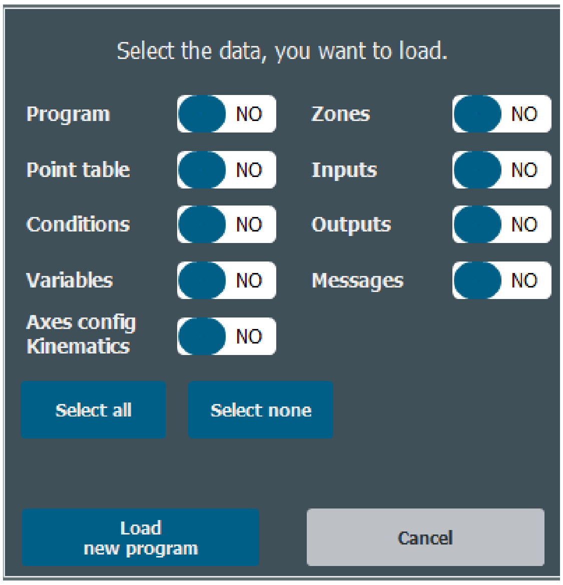
No compatibility between SKO V1.0 and SCP V1.0 programs

Loading programs from LSCP V1.0 with LSCP V1.0 is not supported and can result in unintended behavior.

Data selection for loading

It is possible to load data parts of the csv-file instead of loading the complete csv-file. For example, it is possible to load only the point table from the csv-file and keep the rest of the data in the data block (cf. Figure 9-3).

Figure 9-3: Load selection of the program manager



Select the data, you want to load.

Program	NO	Zones	NO
Point table	NO	Inputs	NO
Conditions	NO	Outputs	NO
Variables	NO	Messages	NO
Axes config Kinematics	NO		

External editing of the csv-files on PC

You can edit the csv-files with editors on your PC and transfer them to a USB flash drive. From USB flash drive you can load the programs directly to the PLC or to the HMI.

NOTE

You should not change the format of the csv-file with an external editor. All headers and separators must be kept, and no additional characters should be added.

⚠ **CAUTION**

The program, points and configuration can be changed in external editors.

After loading of the changed csv-file, the axes could execute unintentional movements.

External changed programs should be tested in single-step mode with a low override.

10 Appendix

10.1 Service and support

Industry Online Support

Do you have any questions or need assistance?

Siemens Industry Online Support offers round the clock access to our entire service and support know-how and portfolio.

The Industry Online Support is the central address for information about our products, solutions and services.

Product information, manuals, downloads, FAQs, application examples and videos – all information is accessible with just a few mouse clicks:

support.industry.siemens.com

Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-made offers
– ranging from basic support to individual support contracts. Please send queries to Technical Support via Web form:

siemens.com/SupportRequest

SITRAIN – Digital Industry Academy

We support you with our globally available training courses for industry with practical experience, innovative learning methods and a concept that’s tailored to the customer’s specific needs.

For more information on our offered trainings and courses, as well as their locations and dates, refer to our web page:

siemens.com/sitrain

Service offer

Our range of services includes the following:

- Plant data services
- Spare parts services
- Repair services
- On-site and maintenance services
- Retrofitting and modernization services
- Service programs and contracts

You can find detailed information on our range of services in the service catalog web page:

support.industry.siemens.com/cs/sc

Industry Online Support app

You will receive optimum support wherever you are with the "Siemens Industry Online Support" app. The app is available for iOS and Android:

support.industry.siemens.com/cs/ww/en/sc/2067

10.2 Industry Mall

The Siemens Industry Mall is the platform on which the entire siemens Industry product portfolio is accessible. From the selection of products to the order and the delivery tracking, the Industry Mall enables the complete purchasing processing – directly and independently of time and location:
mall.industry.siemens.com

10.3 Application support

Siemens AG
Digital Factory Division
Factory Automation
Production Machines
DF FA PMA APC
Frauenauracher Str. 80
91056 Erlangen, Germany

mailto: tech.team.motioncontrol@siemens.com

10.4 Links and literature

Table 10-1: Links and literature

Nr.	Thema
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to this entry page of this application example https://support.industry.siemens.com/cs/ww/en/view/Entry-ID

10.5 Change documentation

Table 10-2: Change documentation

Version	Date	Modifications
V1.0	12/2020	First version
V2.0	03/2023	Update to LSCP V1.0